2024

Efficient and Effective Learning of Foundational Large Multi-Modal Models

Yiren Jian
Dartmouth College, yiren.jian.gr@dartmouth.edu

EFFICIENT AND EFFECTIVE LEARNING OF FOUNDATIONAL LARGE MULTI-MODAL MODELS

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

in

Computer Science

by

Yiren Jian

DARTMOUTH COLLEGE

Hanover, New Hampshire

January 2024

Examining Committee:

Soroush Vosoughi, Chair

SouYoung Jin

V.S. Subrahmanian

Yaoqing Yang

F. Jon Kull, Ph.D.
Dean of the Guarini School of Graduate and Advanced Studies

Abstract

The investigation of large multi-modal models (LMMs) has emerged as a focal point within the Deep Learning community, showcasing its prominence in contemporary research. LMMs exhibit the capacity to take data from diverse modalities, enabling them to execute a myriad of tasks by leveraging complementary information for enhanced predictive capabilities. The learning process of LMMs is bifurcated into two crucial stages: the computationally intensive pre-training stage, aimed at acquiring general representations from web-scale noisy data, and the subsequent fine-tuning stage, focusing on adapting pre-trained models to specific tasks.

Traditionally, the pre-training of foundational LMMs has been considered a privilege limited to research labs with abundant computational resources. In this thesis, we propose a new method for the effective pre-training of foundational vision-language models (VLMs). This involves mitigating the data demands by employing off-the-shelf frozen large language models (LLMs) through a specialized pre-training process. Additionally, we introduce an efficient VLM pre-training method that reduces redundancy in modality projection. Through our novel approach, the data requirements for training LLMs are substantially reduced from 129 million to 4 million instances, and the associated training budget can be curtailed to 1/10 without perceptible decreases in performance.

Furthermore, we present a straightforward yet potent temporal fusion mechanism for adapting pre-trained image-language models to downstream video tasks. Our video

captioning models achieve competitive performance against state-of-the-art benchmarks without extensive pre-training on video-text datasets. Beyond the established domains of multi-modal research in computer vision and natural language processing, our research extends into the realm of bioinformatics by investigating protein-RNA models for multi-modal learning. Our findings demonstrate that pre-trained protein models encapsulate information about biological structures that can be shared with RNAs. Given the limited number of experimentally solved of RNA structures, our discovery opens avenues for novel research directions in transfer learning between proteins and RNAs.

Finally, we employ physical augmented simulations to train a T-cell-peptide model highlights that integrating such simulations in machine learning significantly enhances model training, especially with limited labeled data. This underscores the potential of merging simulations with machine learning, providing a valuable strategy for advancing LMMs training in the biological domain.

Preface

I extend my deepest gratitude to my advisor, Soroush Vosoughi, for his unwavering support throughout my Ph.D. research. Having you as my advisor has been a privilege, and I appreciate the guidance, unparalleled freedom, and encouragement to explore the frontiers of machine learning research.

I also express my thanks to SouYoung Jin, V.S. Subrahmanian, and Yaoqing Yang, who served on my thesis committee. Their dedicated reading of my thesis and invaluable comments and feedback have greatly enriched the quality of my work.

A special acknowledgment goes to Lorenzo Torresani for his guidance during my initial three years at Dartmouth College. I am grateful to my friends and collaborators at Dartmouth College and other institutions; your contributions have been instrumental in my academic journey.

Lastly, heartfelt thanks to my family and friends for their unwavering support and encouragement throughout my Ph.D. studies and life. Your presence has made this journey more meaningful, and I am truly grateful for your constant support.

Contents

Chapter 1

Introduction

Over the past decade, significant progress in Deep Learning research has yielded notable achievements in diverse domains, including image classification, image segmentation, action recognition, and language modeling. While these models exhibit proficient performance within specific tasks through training on extensive, domain-specific datasets, contemporary investigations have pivoted towards the development of models endowed with the capability to interpret information across various modalities, such as vision, language, and audio.

Moreover, recognizing the potential for improved model predictions, recent studies advocate for training models that seamlessly integrate information from disparate modalities. For instance, in the context of an online conference, presenting a video to the model facilitates enhanced summarization by concurrently considering both visual contents (depicting human activities) and auditory cues (capturing conversational dynamics). This integration of complementary modalities is posited to contribute to more informed decision-making processes.

Additionally, the pursuit of multi-modal learning endeavors to emulate the human capacity to obtain knowledge from diverse sources. By fostering the acquisition of abilities akin to human sensory and cognitive functions, these models aim to transcend

unimodal constraints, exemplifying a convergence towards a holistic understanding of information encompassing both perception and expression.

The burgeoning interest in computer vision and natural language processing has propelled significant advancements in the domain of multi-modal learning, particularly in the development of vision-language models. The prevailing paradigm governing these models unfolds across two stages:

- **Pre-training Stage**: This initial phase entails the model's pre-training using extensive web-scale datasets, facilitating the acquisition of comprehensive knowledge encompassing both vision and language domains. Commonly denoted as "Foundational Models", these pre-trained models serve as the bedrock, capturing intricate patterns and representations inherent in multi-modal data.

- **Fine-tuning Stage**: Subsequent to pre-training, the foundational models undergo fine-tuning to cater to specific tasks. Notably, certain scenarios obviate the need for fine-tuning, allowing models to generate predictions through in-context learning. This stage plays a pivotal role in tailoring the model's capabilities to task-specific requirements.

In the subsequent sections, we will discuss an in-depth exploration of these two training stages. This thesis introduces a novel modality projection module and proposes a novel learning paradigm aimed at augmenting the efficiency of pre-training vision-language models. Additionally, novel fine-tuning modules will be expounded upon, addressing the challenge of adapting pre-trained foundational models to specific tasks, especially in instances characterized by limited training examples. Through these contributions, this research aims to advance the current understanding and efficacy of multi-modal learning within the realm of vision-language models.

Section 1.1

Pre-training of Foundational LMMs

The contemporary trajectory of pre-training large multi-modal models finds its roots in seminal works, such as CLIP [154]. CLIP is a dual encoder model characterized by a visual encoder designed to process images and a language encoder dedicated to processing textual information. Operating on the principle of image-text pairs, where each pair comprises an image and its corresponding caption, CLIP takes the encoding of both modalities and seeks to maximize the cosine similarity between the resulting encoded representations. Through extensive training on vast datasets consisting of hundreds of millions of web-crawled image-text pairs, CLIP has demonstrated remarkable zero-shot capabilities, particularly in the image classification, as well as image-text and text-image retrievals. This noteworthy proficiency is ascribed to CLIP's utilization of rich semantic information embedded in language as a form of supervision, diverging from the conventional approach reliant on one-hot hard labels. The departure from traditional training methods underscores the effectiveness of leveraging nuanced semantic relationships present in textual data, contributing to CLIP's impressive performance in tasks that demand cross-modal understanding and generalization.

While CLIP has demonstrated proficiency in image classification and retrieval tasks, it encounters challenges in more intricate tasks such as fine-grained modeling between images and text, leading to limitations in advanced benchmarks like visual entailment and closed-set Visual Question Answering (VQA). Recognizing these limitations, successive models, exemplified by ALBEF [113], have emerged with innovative architectures aimed at addressing these nuanced shortcomings. ALBEF introduces a novel design that amalgamates the foundational principles of CLIP with

the incorporation of a cross-attention module. This module facilitates the infusion of visual features into the language model, allowing a deeper interaction between the two modalities. This strategic integration of visual and linguistic domains results in an enhanced capacity for modeling fine-grained relationships, thereby overcoming the challenges encountered by CLIP in tasks demanding deeper interactions between images and textual content.

In more recent developments, the success of pre-trained language generative models [242] has spurred the evolution of visual-language generative models, designed to produce open-ended free text conditioned on visual input. CoCa [228] stands out as one of the pioneering works in this domain, drawing inspiration from the architecture of ALBEF. CoCa's pre-training involves leveraging a massive dataset comprising billions of image-text pairs, underscoring the importance of scale in achieving its success. The efficacy of CoCa can be attributed, in part, to the utilization of thousands of accelerator units (e.g., TPUs) during the extensive training process. It is noteworthy, however, that reproducing such results may pose challenges for research laboratories with limited computational resources. Despite this constraint, CoCa attained a state-of-the-art status in various visual-language generation benchmarks at the time of its introduction.

The escalating computational demands associated with models such as CoCa have prompted researchers to explore alternative approaches, leading to a shift towards leveraging off-the-shelf pre-trained unimodal models. This entails the use of pre-trained visual encoders, exemplified by architectures like Vision Transformer (ViT), in conjunction with frozen LLMs. The adoption of frozen LLMs offers several advantages in the pursuit of cost-effective and efficient model development: (1) Preservation of Language Ability: LLMs have demonstrated exceptional proficiency in language understanding and generation. Freezing these models at their pre-trained state maximally retains

the acquired language abilities, ensuring that the model capitalizes on the wealth of linguistic knowledge encoded in the pre-training phase. (2) Computational Savings: The training of large language models incurs a substantial computational cost. Leveraging the pre-trained weights of LLMs obviates the need for retraining from scratch, resulting in significant computational savings. This approach aligns with the principles of resource efficiency, making it particularly appealing for researchers operating within constraints of computational resources. (3) Mitigation of Catastrophic Forgetting: By refraining from modifying the parameters of the frozen LLMs, the risk of catastrophic forgetting is mitigated. This not only conserves GPU memory but also expedites training convergence. The stability introduced by using pre-trained and frozen LLMs facilitates a smoother and more efficient training process.

A seminal contribution within the paradigm of utilizing frozen ViTs and LLMs is exemplified by BLIP-2 [115]. In the architecture of BLIP-2, a Query-Transformer (Q-former) assumes the role of the modality projector, establishing a connection between the ViT and the LLM. The Q-former is trained to query a fixed number (e.g., 32) of tokens from the pool of visual features within the ViT, typically comprising 256 tokens. Subsequently, the queried visual features are transmitted to the LLM as a soft-prompt, guiding the LLM to generate the corresponding textual output. Notably, BLIP-2 has showcased outstanding performance in diverse tasks, including zero-shot VQA, image-text and text-image retrievals, and fine-tuned image captioning, surpassing the capabilities of CoCa. Remarkably, the training of BLIP-2 can be executed on an 8 A100-80G server (or 16 A100-40G) within approximately 10 days. This efficiency underscores the feasibility of large-scale pre-training for VLMs, even for research laboratories with moderate computational resources. As a result, BLIP-2 has emerged as a widely adopted foundational model for numerous downstream tasks, consolidating its position as a key reference within the evolving landscape of VLMs.

While BLIP-2 has notably advanced the feasibility of training effective VLMs without massive GPUs, it is pertinent to acknowledge that its training regimen still necessitates a huge dataset of 129 million image-text pairs. Furthermore, the computational demands, albeit reduced compared to some counterparts (e.g., CoCa), remain a consideration, particularly for research laboratories with limited resources. This realization underscores the need for new approaches to mitigate the challenges posed by both data demand and computational demand. Within the context of this thesis, two novel contributions, BLIText [84] and SimVLG [86], are introduced to specifically address these challenges:

- **BLIText** addresses the challenge of high data demand by introducing a novel training strategy. In its methodology, the model undergoes an initial pre-training phase exclusively on a LLM. This phase is crucial for identifying the optimal visual prompt that maximally aligns with the capabilities of the LLM. Subsequently, the visual features are aligned with this identified prompt. This pre-training approach significantly mitigates the data requirements for training traditional VLMs. By emphasizing the synergy between language understanding and visual features through this refined pre-training process, BLIText achieves notable reductions in the volume of required image-text pairs for subsequent VLM training.

- **SimVLG** targets the computational demands inherent in training Visual Language Models. Central to its approach is the introduction of a novel modality connector termed the Token-Merging Transformer. This component plays a pivotal role in accelerating the convergence of modality alignment, thereby substantially reducing the number of training iterations required. The Token-Merging Transformer acts as an efficient bridge between visual and linguistic modalities, streamlining the learning process and enhancing the computational

efficiency of VLM training. SimVLG's contribution lies in its ability to make VLM training more accessible and feasible on setups with limited computational resources, ultimately broadening the applicability of these models in diverse research settings.

Section 1.2

Adapting Foundational LMMs to Downstream Tasks

Upon acquiring foundational knowledge through pre-training on extensive and readily available web-crawled datasets, the subsequent refinement of models becomes imperative through fine-tuning on meticulously annotated datasets tailored to specific downstream tasks. Traditional fine-tuning methodologies involve the comprehensive adjustment of all parameters within the network in an end-to-end fashion. However, in the context of LLMs and LMMs, where the model's parameters scale to the order of billions, a judicious approach is essential. This necessitates a consideration of several critical issues. Notably, VLMs typically consist of three integral modules: a visual encoder, modality projection, and the LLM. It is important to determine which components of the model warrant fine-tuning, given the intricacies posed by the vast parameter space.

BLIP-2's findings underscore the advantageous impact of fine-tuning ViT models on subsequent image captioning tasks. In the area of instruction-based learning, InstructBLIP [30] adopts a targeted approach by exclusively fine-tuning the vision-language connector(i.e., Query-Transformer), while maintaining the ViT and the LLM in a frozen state. Conversely, alternative methods introduced by other researchers involve the integration of LoRA (Low-Rank Adaptation) modules into LLMs. This augmentation not only enhances the models' capacity but also introduces the capability

to dynamically influence and guide the output of the LLMs.

A frequently investigated scenario involves the adaptation of foundational VLMs, initially pre-trained on image-text pairs, to the domain of video-language tasks. VideoCoCa [221] extends this paradigm by subjecting CoCa to further pre-training utilizing billion-scale video-text pairs. Notably, VideoCoCa strategically fine-tunes the attentive pooler, directing attention primarily towards spatial features while overlooking temporal modeling aspects. In contrast, Video-LLaMA [236] introduces a temporal Q-former to imbue VLMs with temporal modeling capabilities. However, this enhancement comes with the prerequisite of a substantial corpus of video-text pairs to facilitate the re-alignment of ViT and LLMs. Similarly, the VideoChat [116] approach incorporates UniFormer modules into ViT to address temporal modeling requirements. Regrettably, the incorporation of these modules disrupts the integrity of the well-trained VLMs, presenting a trade-off between temporal modeling efficacy and model stability.

In this thesis, we propose Attentive Temporal Token Merging Modulesfor ViTs, aiming to imbue ViTs with temporal modeling capabilities while maintaining the integrity of well-pre-trained VLMs. This approach eliminates the need to re-align ViTs and LLMs without relying on massive video-text pairs.

Section 1.3

Main Contributions and Outlines

In this dissertation, we present a series of contributions aimed at advancing the field of vision-language learning and multi-modal learning, addressing challenges related to data efficiency, training resource constraints, and domain expansion. Our primary focus lies in the development of novel models and methodologies that significantly enhance the efficacy and accessibility of VLMs.

- Firstly, we introduce BLIText, a data-efficient pre-trained VLM that notably reduces the demand for pre-training data from 129 million to 4 million instances. This innovation represents a substantial stride toward mitigating the resource-intensive nature of traditional pre-training processes.

- To facilitate the training of large VLMs in research environments with limited resources (i.e., GPU hours), we propose SimVLG, a training-efficient model. This contribution aims to democratize the training of formidable VLMs, making them more accessible to research labs operating under resource constraints.

- Furthermore, we present SimVLG-video, an extension of SimVLG tailored for video-language tasks. This adaptation incorporates a novel Temporal Token Merging technique, enhancing the model's effectiveness in handling temporal aspects inherent in video data.

- In the field of vision models, we introduce LabelHalluc [77], a method designed to maximize the utility of base datasets for effective fine-tuning. This approach seeks to optimize the transferability of pre-trained vision models to specific tasks, thereby reducing the reliance on task-specific data.

- For language models, we introduce LM-SupCon [80], utilizing contrastive learning to mitigate overfitting challenges during adaptation. This approach improves the generalization capabilities of language models, especially in situations with limited adaptation datasets. Given that contemporary VLMs typically employ frozen LLMs as decoders, the implications of LM-SupCon extend significantly into the VLM domain.

- Lastly, we extend our multi-modal exploration beyond traditional vision-language domains into the field of bioinformatics. Demonstrating the versatility of pre-trained protein models, we showcase their effectiveness as multi-modal biological

models capable of accurately encode information of RNA structures [85]. Additionally, akin to traditional VLMs handling image-text pairs, we construct a T-cell-peptide interaction model. This model holds significance in predictions of reactions in human immunity [83].

Through these contributions, our research not only contributes to the advancement of vision-language models and multi-modal learning but also underscores the broader applicability of such models in diverse domains.

Chapter 2

Related Work

Section 2.1

Vision-Language Learning

In Chapter 3 and Chapter 4, we introduce two vision-language models, BLIText [84] and SimVLG [86] which significant reduce the demand for training data and computation, respectively. The following covers the related work on vision-language modeling that is investigated/discussed in "Bootstrapping Vision-Language Learning with Decoupled Language Pre-training" [84] and "SimVLG: Simple and Efficient Pretraining of Visual Language Generative Models" [86].

The landscape of end-to-end vision-language pre-training models generally falls into two overarching categories: dual-encoder and fusion-encoder models. In the dual-encoder paradigm, distinct networks are dedicated to processing visual and linguistic information. The interaction between modalities is achieved through the computation of the dot product between corresponding features, as seen in models like CLIP [154]. Notably, the effectiveness of dual-encoder models shines in tasks involving image-text retrieval, owing to the efficient computation of vector dot-products facilitated by feature caching. Nevertheless, their efficacy faces limitations in tasks such as VQA,

captioning, and visual reasoning, primarily attributed to the deficiency in establishing fine-grained alignment between the two modalities.

Dual-encoder architectures utilize distinct neural networks dedicated to processing visual and linguistic information. The interaction between these modalities is achieved by computing the dot product between corresponding visual and linguistic features, as demonstrated by CLIP [154]. The efficiency of dual-encoder models in image-text retrieval tasks is notable, attributed to the streamlined computation of vector dot-products facilitated by feature caching. Despite their effectiveness in this domain, their performance in tasks such as VQA, captioning, and visual reasoning is constrained due to the absence of nuanced alignment between the two modalities.

Fusion-encoder models, exemplified by ALBEF [113], VLMo [9], and CoCa [228], incorporate novel fusion-Transformer layers to capture intricate interactions between vision and language, augmenting traditional vision and language encoders. These models adopt various design approaches, such as concatenating visual and linguistic features prior to inputting them into a self-attentive Transformer [9, 25, 27, 53, 73, 74, 94, 98, 117, 119, 121, 175, 189, 204, 206, 207, 210, 220, 225, 241], or employing cross-attention mechanisms between vision and language encoders to compute fused features [3, 42, 43, 111, 113, 114, 125, 128, 130, 195, 218]. The vision encoder exhibits diversity, ranging from straightforward linear embeddings [98] and ConvNets [73, 74, 94, 175, 206, 210, 225] to Transformer architectures [9, 42, 43, 113, 114, 204, 207, 220], employing offline pre-trained object detectors like Faster-RCNN [25, 27, 53, 117, 119, 121, 189, 241], or integrating ensemble models [127]. The language encoder may be initialized with a BERT-based [96] model or serve as part of a fusion-Transformer [9, 42, 43, 207, 231]. During pre-training, most methods employ three primary losses: image-text contrastive (ITC) loss, image-text matching (ITM) loss, and either mask language modeling (MLM) loss or language generation (ITG) loss. While

fusion-encoder models exhibit notable success in VQA and captioning tasks, their efficiency in retrieval tasks is comparatively lower. A comprehensive overview of recent advancements in vision-language pre-training is available in [54].

Language models trained on extensive text corpora exhibit remarkable proficiency in language generation tasks. Consequently, integrating these large, pre-trained LLMs into Vision-Language models proves advantageous, especially in tasks such as VQA and captioning. Flamingo [3] incorporates visual signals into each layer of a large frozen LLM through cross-attention mechanisms. In contrast, Frozen [199] fine-tunes the image encoder, aligning visual features as soft prompts input into the frozen language model. More recently, BLIP-2 [115] introduced the Q-former, a vision-to-language adaptation module used in conjunction with a frozen ViT [41] and an LLM. BLIP-2 employs a two-stage training process to address challenges in learning visual-language alignment. In the initial stage, the Q-former is optimized to extract beneficial visual features, utilizing Image-Text Contrastive (ITC), Image-Text Matching (ITM), and Image-Text Generation (ITG) losses. Subsequently, in the second stage, all three modules (ViT, Q-former, and LLM) undergo end-to-end training, with only the parameters in the Q-former updated. Despite being trained on a dataset comprising 129 million image-text pairs and using relatively modest computational resources, BLIP-2 exhibits competitive performance across various benchmarks. Additionally, concurrent work on the visual chat-bot X-LLM [21] follows a similar architectural design philosophy to BLIP-2, further reinforcing the efficacy of this approach.

In addition to leveraging off-the-shelf pre-trained vision encoders like ViT and Faster-RCNN [59, 162] and language models, there is a compelling interest in exploring how unimodal training can contribute to the enhancement of multi-modal models. VLMo [9] exemplifies the advantages of a stage-wise pre-training approach using image-only and text-only data for their proposed model architecture. A different

perspective is presented by Li et al. [118], who suggest employing object tags from detectors as anchor points to establish connections between unpaired images and text. Similarly, Zhou et al. [247] create pseudo-image-text pairs through an image-text retrieval alignment strategy. In video-language models, image-text pairs are utilized by repeating images to generate static videos, thereby constructing auxiliary paired datasets for pre-training. Furthermore, Jian et al. [82] demonstrated that contrastive visual learning not only benefits visual tasks but also contributes to the improvement of contrastive sentence embeddings, highlighting the potential impact of unimodal training on tasks primarily focused on language.

Section 2.2

Efficient and Effective Adaptation of LMMs

In Chapter 5, we introduce SimVLG-video [86] as an effective method for adapting the well pre-trained image-based model SimVLG to video understanding tasks. The following covers the related work on vision-language modeling that is investigated/discussed in "SimVLG: Simple and Efficient Pretraining of Visual Language Generative Models" [86].

Image-Language Models to Video-Language Models While many models designed for image-text tasks can be adapted for video-text applications using simple feature pooling, as demonstrated by VideoCoCa [221], specialized models incorporating temporal dynamics have emerged. Expanding upon the groundwork laid by BLIP-2, Video-LLaMA [236] enhances its architecture by introducing additional temporal Q-former layers positioned between the spatial Q-former and the LLM components of the original BLIP-2 model. Following the inspiration from BLIP-2, recent works such as VideoChat [116], PandaGPT [190], Valley [132], and Video-ChatGPT [139] have

adopted the use of frozen LLMs in their video-language models.

Token Merging Token Merging (ToMe) [13] seeks to enhance the inference speed of pre-trained Vision Transformers (ViTs) without necessitating re-training. In this approach, tokens are partitioned into two sets at each Transformer layer and then merged based on their similarity. This process effectively reduces the overall token count, leading to an acceleration in inference speed. Notably, ToMe achieves this without compromising the quality of classification and generation tasks.

In Chapter 5, we adapt ToMe to compress the visual features utilized as language prompts in the LLM. Our approach involves incorporating ToMe capabilities into a standard Transformer, resulting in a novel model termed TomeFormer. This model functions as a robust bridge between the visual and language domains, retaining semantic richness while reducing the token count. Crucially, the integration of ToMe in TomeFormer does not introduce any extra parameters. Drawing inspiration from spatial ToMe, we introduce a new variant called soft temporal ToMe within the vision encoder, thereby enhancing our image-text models with temporal modeling capabilities.

In Chapter 5, we also propose Label Hallucination [77] as a general effective approach to adapt the pre-trained model, given a few examples (i.e., few-shot learning). The high-level idea rooted in Label Hallucination is Transfer Learning plus pseudo-labeling of base examples, using the label space of novel few-shot datasets. The following covers the related work on vision-language modeling that is investigated/discussed in "Label Hallucination for Few-shot Classification" [77].

Pseudo-labeling or self-training [108, 213] involves initially labeling the unlabeled dataset using the model itself and subsequently re-training the model with both the labeled and pseudo-labeled datasets. This technique has exhibited significant success

in semi-supervised learning scenarios [11, 184, 229], particularly when pseudo-labeling is applied to a large unlabeled dataset with classes that overlap with the labeled set (i.e., the unlabeled and labeled datasets share similar distributions). Algorithms are often designed to filter out low-confidence pseudo-labeled examples. The recent work by Pham et al. [150], combining gradient-based meta-learning with pseudo-labeling, has achieved a state-of-the-art performance on the ImageNet benchmark [36]. This approach has also been extended to semi-supervised few-shot learning [106, 120, 209], and in the application of semantic segmentation [76].

Furthermore, BLIP [114] has demonstrated that pseudo-labeling noisy web-crawled image-text pairs is an effective method for learning vision-language models.

Transfer learning stands as the predominant approach for training effective models in various tasks [40], especially when labeled examples are limited. Recent baseline methods [24, 38] have demonstrated competitive performances in few-shot classification by employing pretraining on a base training set and subsequent finetuning on the support set from each episode. RFS [196] has surpassed advanced meta-learning methods of its time by adopting a fixed embedding model followed by linear regression. The efficacy of transfer learning methods heavily relies on the quality of pretrained feature embeddings. To obtain more generalized embeddings of examples, SKD [155] proposes the inclusion of a rotational self-supervised loss during the pretraining stage. Invariant and Equivariant Representations (IER) [165] delves into contrastive learning during the embedding learning process.

Chapter 3

VLM Pre-training with Less Data

In this Chapter, we present BLIText (Bootstrapping Vision-Language Learning with Decoupled Language Pre-training [84]). This work has been published in NeurIPS 2023.

Section 3.1

Overview

We present a novel methodology aimed at optimizing the application of frozen large language models (LLMs) for resource-intensive vision-language (VL) pre-training. The current paradigm uses visual features as prompts to guide language models, with a focus on determining the most relevant visual features for corresponding text. Our approach diverges by concentrating on the language component, specifically identifying the optimal prompts to align with visual features. We introduce the Prompt-Transformer (P-Former), a model that predicts these ideal prompts, which is trained exclusively on linguistic data, bypassing the need for image-text pairings. This strategy subtly bifurcates the end-to-end VL training process into an additional, separate stage. Our experiments reveal that our framework significantly enhances the performance of a robust image-to-text baseline (BLIP-2), and effectively narrows the

performance gap between models trained with either 4M or 129M image-text pairs. Importantly, our framework is modality-agnostic and flexible in terms of architectural design, as validated by its successful application in a video learning task using varied base modules.

Section 3.2

Motivation

The field of vision-language (VL) learning seeks to create AI systems that mimic human cognition, processing the world through multi-modal inputs. Core research areas in VL include visual-question-answering (VQA), image captioning, image-text retrieval, and visual reasoning. VL learning began with task-specific learning [5, 217] and has since progressed to large-scale image-text pre-training paired with task-specific fine-tuning [154]. Furthermore, contemporary studies have begun exploring the use of off-the-shelf frozen pre-trained large language models (LLMs) in VL models [3, 86, 115, 199], which have delivered impressive results in language generation tasks such as VQA and image captioning.

Present VL models utilizing frozen LLMs are characterized by shared design elements: visual encoders, visual-to-language modules, and frozen LLMs. Except for Flamingo [3], which employs a visual signal at each layer of the frozen LLM via gated cross-attention, the majority of works [21, 115, 126, 136, 199] feed aligned visual features as soft language prompts [110] into the frozen LLMs (see Figure 3.1 ***left***). The models are then trained end-to-end with an image-conditioned language generation loss using large-scale image-text pairs. This conceptually simple and implementation-wise straightforward design has proven effective. BLIP-2 [115] demonstrates that decoupling the end-to-end training into two stages is crucial for state-of-the-art results. The second stage of training involves standard end-to-end learning, while the first stage of

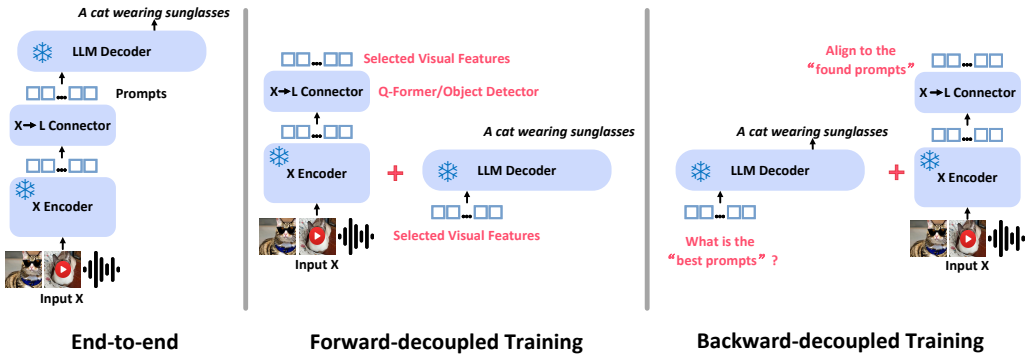

Figure 3.1: ***left:*** End-to-end training of X-to-language models (where X can be images, videos, or audio), in which aligned input features are provided as prompts to LLMs. Examples include Frozen [110] and ClipCap [136]. ***middle:*** "Forward-decoupled training" as demonstrated in BLIP-2 [115] and X-LLM [21]. For instance, in BLIP-2, the Q-Former is first trained to extract relevant features from the image encoder, and then the selected features are used as prompts for LLM for end-to-end learning. ***right:*** We propose "backward-decoupled training", which initially identifies the "reference prompt" for the LLM to generate the target text, followed by mapping input features to the "reference prompt".

training of BLIP-2 utilizes a learnable module (called Query-Transformer/Q-Former) to selectively choose/query visual features relevant to the corresponding text. This reduces 256 features of an entire image to the 32 most relevant visual features that will be sent into the following parts of the model. Stage 1 of BLIP-2 can be viewed as a refined learnable version of early VL works [5, 121, 241] that use object detectors like Faster-RCNN [59] to select features from regions of objects (objects in images are likely to be mentioned and thus relevant to the accompanying text). We refer to this strategy as "forward-decoupling" since it uses a heuristic to learn/select which useful features are forward-passed into the subsequent model to mitigate challenges in the end-to-end optimization (shown in Figure 3.1 ***middle***).

We provide a novel insight to mitigate the challenges in end-to-end optimization by introducing "backward-decoupling" during back-propagation. For a caption t (e.g., *"a cat wearing sunglasses"*) from VL pre-training dataset $\mathcal{D}_{\text{VL}}$, the optimizer first finds

the optimal continuous prompt p for a fixed decoder LLM D_{language}:

$$p = \underset{p}{\operatorname{argmin}} \mathcal{L}(D_{\text{language}}(p), t) \tag{3.1}$$

before further back-propagating into the vision-to-language module (e.g., Q-Former in BLIP-2, or MLP in ClipCap) and the vision encoder (shown in Figure 3.1 ***right***). We realize that the first stage, optimization of p given D_{language} and t, is purely linguistic and does not restrict the learning text examples from $\mathcal{D}_{\text{VL}}$. Thus, we propose to learn this part independently with the available sentence dataset.

While it's not feasible to learn individual prompts p for each sentence t due to the infinite number of possible sentences, we propose to parameterize prompt p by a Prompting-Transformer (P-Former): $p = E_{\text{P-Former}}(t)$. This effectively transforms the learning of p given D_{language} and t into learning $E_{\text{P-Former}}$ by

$$\underset{E_{\text{P-Former}}}{\operatorname{argmin}} \mathcal{L}(D_{\text{language}}(E_{\text{P-Former}}(t)), t) \tag{3.2}$$

Essentially, this is an autoencoder with the causal LLM D_{language} as the decoder. As for P-Former, we use a bidirectional Transformer and the [CLS] representation as the bottleneck. Besides the reconstruction loss, we add a contrastive loss to discriminate each sample. Such a design makes $E_{\text{P-Former}}$ a semantic sentence embedding model like SimCSE [56] (i.e., semantically similar sentences have similar representations). Once $E_{\text{P-Former}}$ is learned, $p = E_{\text{P-Former}}(t)$ will be the "reference prompt" for LLM D_{language} to generate t auto-regressively. The training overview and P-Former details are shown in Figure 3.2.

Returning to the VL pre-training, we add a complementary loss to minimize the distance between aligned visual features (being used as language prompts) and the "reference prompt" given by P-Former. We expect this to improve the VL pre-training

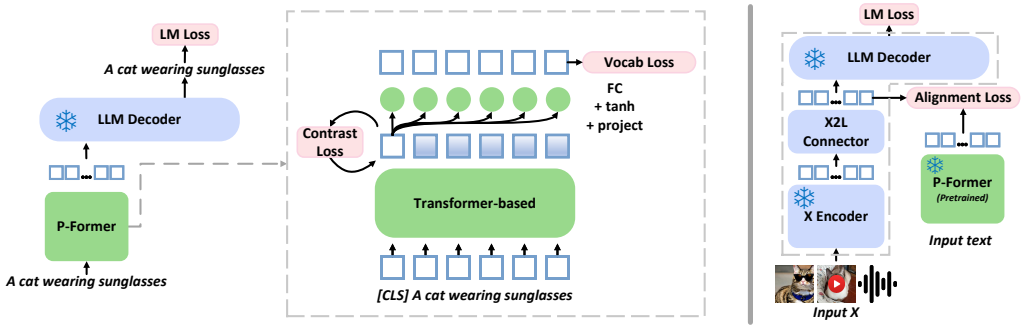

Figure 3.2: Overview of P-Former. ***left:*** The P-Former training resembles an autoencoder, with the bidirectional P-Former as the encoder and a causal LLM (frozen) as the decoder. The objective is to reconstruct input text auto-regressively. The [CLS] representation serves as sentence embeddings, which are projected back to the length of prompts. The contrastive loss at [CLS] mirrors the training of SimCSE [56]. A regularization vocabulary loss is utilized to encourage the prompts to be close to the vocabulary embeddings. ***right:*** Overview of bootstrapping VL pre-training with the trained P-Former. The alignment loss introduced by P-Former is agnostic to input modalities, encoders, and X-to-language modules (i.e., modules within the dashed box can be flexible). P-Former is only used during training and not during inference.

in two ways: (1) We further decouple the VL learning into another stage, as Li et al. [115] suggest that multi-stage training is important to mitigate alignment challenges. (2) A semantically rich space is learned for aligned visual features/prompts by a SimCSE design for our P-Former trained with the unimodal sentence dataset (i.e., semantically similar images are encouraged to align to "reference prompts" with close representations).

Our proposed framework only adds a learning objective on tensors feeding into LLMs as prompts (a.k.a images/multi-modalities as foreign languages [21, 207]). Therefore, our method is agnostic to the input modalities, X encoders, and X-to-language modules (where X can be images, videos, and audio). This could be especially salient for videos, which have much less high-quality paired data [54] compared to image-text pairs. And because P-Former is only trained with the LLM, there is no need to re-train the P-Former for different modalities.

In our experiments, we take BLIP-2 as an example and show that our proposed

framework improves this latest VL method by great margins in various benchmarks of VQA and image captioning. Also, we demonstrate its effectiveness in other modalities (i.e., video) using different vision-to-language modules (i.e., plain Transformer over Q-Former).

We anticipate a growing body of future work within the paradigm of "images/multi-modalities as language prompts with frozen LLMs" due to its simplicity and effectiveness, as demonstrated by BLIP-2. For example, a concurrent work X-LLM [21] extends BLIP-2 from images to videos/speech with more advanced LLMs, augmenting BLIP-2's vision-to-language module Q-Former with Adapters. Because our proposed method is agnostic to input modalities, encoders, and X-to-language modules, it should seamlessly apply to future work within this paradigm of "images/multi-modalities as language prompts with frozen LLMs".

Section 3.3

Technical Approach

Problem formulation Given an image-text dataset $\{I, t\} \in \mathcal{D}_{\text{VL}}$ and a unimodal language dataset composed purely of sentences $\{t\} \in \mathcal{D}_{\text{L}}$, our objective is to optimize the pre-training of a vision-language (VL) model. This model consists of a pre-trained vision encoder E_{vision}, a vision-to-language adaptation module $\underset{\text{V}\to\text{L}}{\Theta}$, and a frozen pre-trained language decoder D_{language}. The goal is to minimize the image-conditioned language generation loss, given that the vision encoder E_{vision} is also frozen:

$$\underset{\underset{\text{V}\to\text{L}}{\Theta}}{\operatorname{argmin}} \mathcal{L}_{\text{CrossEntropy}}(D_{\text{language}}(\underset{\text{V}\to\text{L}}{\Theta}(E_{\text{vision}}(I))), t) \tag{3.3}$$

As Li et al. [115] have noted, end-to-end optimization of Equation 3.3, visualized in Figure 3.1 ***left***, can sometimes lead to catastrophic forgetting in LLMs.

3.3.1. Backward-decoupling and soft prompt pre-training

Let's denote the adapted visual features as $p = \underset{\mathrm{V}\to\mathrm{L}}{\Theta}(E_{\text{vision}}(I))$, which serve as soft prompts for the LLM D_{language}. During the optimization, Equation 3.3 can be decomposed into two parts, visualized in Figure 3.1 ***right***:

$$\underset{p}{\operatorname{argmin}}\, \mathcal{L}_{\text{CrossEntropy}}(D_{\text{language}}(p), t) \tag{3.4}$$

$$\underset{\underset{\mathrm{V}\to\mathrm{L}}{\Theta}}{\operatorname{argmin}}\, \mathcal{L}_{\text{MSE}}(\underset{\mathrm{V}\to\mathrm{L}}{\Theta}(E_{\text{vision}}(I)), p) \tag{3.5}$$

Equation 3.4 essentially asks *"What is the optimal soft prompt p that enables the auto-regressive language model $D_{language}$ to generate the sentence t."* Like all gradient-based deep learning models, depending on the training dataset, learning p given $\{D_{\text{language}}, t\}$ could lead to different sub-optimal points[1] (a conventional deep learning problem is usually learning D_{language} given $\{p, t\}$). End-to-end learning of Equation 3.3 can only use text t from image-text dataset $\mathcal{D}_{\text{VL}}$ to update its intermediate variable p. However, we observe that the learning of Equation 3.4 involves no image, thus allowing us to leverage abundantly available unimodal sentences in $\mathcal{D}_{\text{L}}$.

Learning p for each t in $\mathcal{D}_{\text{L}}$ without constraint is intractable. Thus, we model p by a bidirectional Transformer $E_{\text{P-Former}}$ (named Prompt-Former, or P-Former) $p = E_{\text{P-Former}}(t)$. Specifically, we use the output [CLS] hidden state of BERT as a compact representation for t and project it back to the token length of p. Equation 3.4 can thus be reformulated as:

$$\underset{E_{\text{P-Former}}}{\operatorname{argmin}}\, \mathcal{L}_{\text{CrossEntropy}}(D_{\text{language}}(E_{\text{P-Former}}(t)), t) \tag{3.6}$$

[1]It can be easily verified that there exist multiple different soft prompts for an LLM to generate the same text auto-regressively. In an extreme example, a prompt with 32 tokens and a prompt with 16 tokens padded with 16 empty tokens (zeros vectors) can be both optimized for a LLM to generate the same text.

In essence, Equation 3.6 describes the training of an autoencoder with the bidirectional P-Former $E_{\text{P-Former}}$ serving as the encoder, and the auto-regressive LLM D_{language} as the decoder. To enhance our model, we include an unsupervised contrastive loss $\mathcal{L}_{\text{contrast}}$, acting on the [CLS] representations of sentences to differentiate distinct instances. This loss, combined with our P-Former design, emulates the training of SimCSE [56], a semantic sentence embedding model (i.e., for semantically similar image-text pairs, the predicted prompts by P-Former should also be close). Furthermore, we introduce a regularization loss $\mathcal{L}_{\text{vocab}}$ to minimize the distance between each token in p and the closest embedding of the LLM's (D_{language}) vocabularies. The final objective becomes:

$$\underset{E_{\text{P-Former}}}{\operatorname{argmin}}(\mathcal{L}_{\text{CrossEntropy}}(D_{\text{language}}(E_{\text{P-Former}}(t)), t) + \mathcal{L}_{\text{contrast}} + \mathcal{L}_{\text{vocab}}) \tag{3.7}$$

A comprehensive view of the P-Former's architecture and learning losses is presented in Figure 3.2 ***left***. We emphasize that the optimization of Equation 3.7 and P-Former training rely only on the text. Upon training the P-Former, Equation 3.5 can be reformulated as:

$$\underset{\substack{\Theta \\ \text{V}\to\text{L}}}{\operatorname{argmin}} \mathcal{L}_{\text{MSE}}(\underset{\text{V}\to\text{L}}{\Theta}(E_{\text{vision}}(I)), E_{\text{P-Former}}(t)) \equiv \underset{\substack{\Theta \\ \text{V}\to\text{L}}}{\operatorname{argmin}} \mathcal{L}_{\text{alignment}} \tag{3.8}$$

This new form, depicted in Fig 3.2 ***right***, minimizes the distance between the aligned visual features and the prompts predicted by the trained P-Former, effectively aligning visual-linguistic representations.

3.3.2. Preliminary: BLIP-2 forward-decoupled training

While our proposed framework is flexible in regards to the specific architecture of $\underset{\text{V}\to\text{L}}{\Theta}$ or the learning strategy deployed, for illustrative purposes, we employ BLIP-2 as a case study to demonstrate the applicability of our approach with state-of-the-art

learning methods, owing to the strong performance and reproducibility of BLIP-2. In the context of BLIP-2, E_{vision} is a ViT-g, $\underset{\text{V}\to\text{L}}{\Theta}$ is referred to as Q-Former, and D_{language} is a $\text{OPT}_{2.7\text{B}}$. BLIP-2 proposes a two-stage pre-training process, with the initial stage involving the pre-training of $\underset{\text{V}\to\text{L}}{\Theta}$ by:

$$\underset{\underset{\text{V}\to\text{L}}{\Theta}}{\operatorname{argmin}}\,\text{ITC}(\underset{\text{V}\to\text{L}}{\Theta}(E_{\text{vision}}(I)),\underset{\text{V}\to\text{L}}{\Theta}(t))+\text{ITM}(\underset{\text{V}\to\text{L}}{\Theta}(E_{\text{vision}}(I),t))+\text{ITG}(\underset{\text{V}\to\text{L}}{\Theta}(E_{\text{vision}}(I),t)) \tag{3.9}$$

This is followed by a second stage that involves end-to-end training of Equation 3.3. The terms ITC, ITM, and ITG in Equation 3.9 are utilized to guide the Q-Former $\underset{\text{V}\to\text{L}}{\Theta}$ in extracting visually relevant features that correspond to the associated captions. We refer to this two-step process in BLIP-2 – first determining the visual features to extract and then incorporating the selected visual features into an end-to-end learning framework – as "forward-decoupled training."

3.3.3. BLIP-2 forward-decoupled training with pre-trained P-Former

We now describe the full training pipeline when integrating our framework with BLIP-2. The first stage of training involves pre-training the Q-Former with Equation 3.9 ($\mathcal{L}_{\text{BLIP2-stage1}} \equiv \text{ITC}+\text{ITM}+\text{ITG}$), supplemented with the alignment loss introduced by the P-Former, as defined in Equation 3.8:

$$\mathcal{L}_{\text{BLIP2-stage1}}+\omega_1\times\mathcal{L}_{\text{alignment}} \tag{3.10}$$

Subsequently, the second stage of training, in line with our approach, involves BLIP-2's stage 2, which is the end-to-end training of Equation 3.3: $\mathcal{L}_{\text{BLIP2-stage2}} \equiv \mathcal{L}(D_{\text{language}}(\underset{\text{V}\to\text{L}}{\Theta}(E_{\text{vision}}(I))),t))$, again enhanced with the alignment loss imparted by

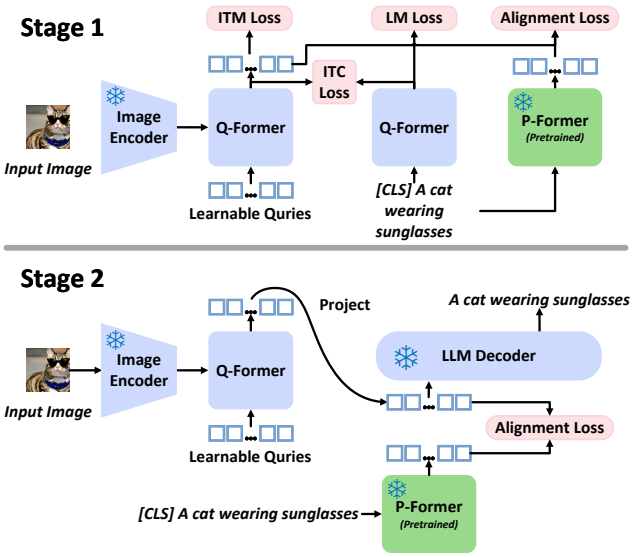

Figure 3.3: An overview of our framework with BLIP-2, which employs a two-stage training process. The green components represent the alignment loss and modules added by us, which do not require gradients. The blue components are part of the original BLIP-2 structure. **P-Former is solely utilized during training and is not required during the inference phase.** Our proposed framework, with P-Former, can be seamlessly applied to any models that leverage prompts as the interface for multi-modal-language communications.

P-Former in Equation 3.8:

$$\mathcal{L}_{\text{BLIP2-stage2}} + \omega_2 \times \mathcal{L}_{\text{alignment}} \tag{3.11}$$

Figure 3.3 provides a schematic representation of the proposed integration of our framework and P-Former with BLIP-2.

3.3.4. Model pre-training

Training dataset We employ a 12M subset of the pseudo-labeled [114] LAION dataset [170], using only the sentences, for pre-training the P-Former. For VL pre-training, we widely adapted academic setting (since academic institutions lack the resources available to industry researchers to use very large datasets) with approximately 4M image-text pairs. This set comprises the MSCOCO-80K [122], VG-100K [101], CC-3M

[173], and SBU-1M [142] datasets.

Pre-training models Our method is universally applicable to any vision-to-text models that utilize prompts as the interface. Owing to its impressive performance and reproducibility, we chose BLIP-2 as the base model for our primary experiments. Thus, for VL pre-training, the image encoder E_{vision} is a ViT-g/14 from EVA-CLIP [47], the LLM decoder D_{language} is an $\text{OPT}_{\text{2.7B}}$ [242], and the vision-to-language adaptation module is a Q-Former [115]. The Q-Former is initialized by BERT-base with 32 learnable queries. Our newly proposed P-Former is a base Transformer initialized by BERT-base.

Pre-training details The P-Former is trained on a system with 3 $\times$ RTX-A6000 (48GB) GPUs, using PyTorch [147]. We trained for five epochs with a linear warm-up and cosine scheduling, using a batch size of 384 (3×128), and AdamW as the optimizer. The initial learning rate is set to $1e^{-4}$, with a minimum learning rate of $1e^{-5}$, a warm-up learning rate of $1e^{-6}$, and 2000 warm-up steps. The VL pre-training is performed on a server equipped with 8 $\times$ RTX-A6000 (48GB) GPUs, using PyTorch. We developed the code based on the LAVIS project [112]. Predominantly, we employed the default configuration files provided by BLIP-2 of LAVIS. Both the stage 1 and stage 2 training ran for 10 epochs with linear warm-up and cosine scheduling, using a batch size of 1024 (8×128), and AdamW as the optimizer. The weight decay is set to 0.05, the initial learning rate is $1e^{-4}$, the minimum learning rate is $1e^{-5}$, and the warm-up learning rate is $1e^{-6}$. The key distinction is that stage 1 and stage 2 incorporate 5000 and 2000 warm-up steps, respectively. We set $\omega_1 = 10$ and $\omega_2 = 100$ while training BLIP-2 $\text{OPT}_{\text{2.7B}}$ with our P-Former.

Computational overhead considerations Incorporating $\mathcal{L}_{\text{alignment}}$ from Equation 3.10 and 3.11 introduces only a minimal computational overhead, attributable to an additional forward pass of the P-Former (Transformer-base) at each iteration.

To illustrate, in our experimental settings using BLIP-2 $\text{OPT}_{2.7\text{B}}$, the training time for stage 1 saw a modest increase from 2,669 minutes to 2,743 minutes. Similarly, for stage 2, the training time increased marginally from 1,862 minutes to 1,880 minutes. Thus, our methodology's overall computational burden remains manageable despite its enhancements (the only additional cost is pre-training of the P-Former, which only needs to be done once for an LLM).

Section 3.4

Experiments

Given the impressive performance and accessibility of the BLIP-2 model, coupled with its open-source nature, we primarily employ it as our base model. We aim to demonstrate how our proposed "backward-decoupling" strategy, along with the learned P-Former, can enhance the baselines across various image-to-text generation benchmarks. In Section 3.4.5, we further extend the applicability of our framework to other modalities, utilizing different base models.

3.4.1. Zero-shot image-to-text generation

We assess the performance of our pre-trained models on zero-shot VQA, encompassing GQA [75], OKVQA [134], and VQAv2 [64], without any task-specific fine-tuning. As per BLIP-2, we append text prompts to visual prompts prior to their processing by the frozen LLM. Both for the baseline BLIP-2 and our model, the text prompt used is "Question: Short answer:". The results, as detailed in Table 3.1, suggest that our proposed framework significantly enhances the zero-shot VQA performance of BLIP-2 trained with 4M image-text pairs. Remarkably, the gap between the BLIP-2 trained with 4M and 129M image-text pairs is largely bridged by our method.

Models	#Pretrain Image-Text	Pretrain Uni-Text	VQAv2 val	VQAv2 test-dev	OK-VQA test	GQA test-dev
FewVLM [87]	9.2M	-	47.7	-	16.5	29.3
Frozen [199]	3M	-	29.6	-	5.9	-
VLKD [29]	3M	-	42.6	44.5	13.3	-
Flamingo3B [3]	1.8B	-	-	49.2	41.2	-
$OPT_{2.7B}$ BLIP-2 [115]	4M	-	46.8	45.6	25.9	30.5
$OPT_{2.7B}$ Ours	4M	✓	<u>52.6</u>	<u>52.2</u>	<u>30.0</u>	<u>34.0</u>
$OPT_{2.7B}$ BLIP-2† [115]	129M	-	**53.5**	**52.3**	**31.7**	**34.6**

Table 3.1: Comparison with different methods on zero-shot VQA †: numbers taken from Li et al. [115].

Models	#Pretrain Image-Text	NoCaps Zero-shot (validation set) in-domain C	in-domain S	near-domain C	near-domain S	out-domain C	out-domain S	overall C	overall S	COCO Fine-tuned Karpathy test B@4	COCO Fine-tuned Karpathy test C
OSCAR [121]	4M	-	-	-	-	-	-	80.9	11.3	37.4	127.8
VinVL [241]	5.7M	103.1	14.2	96.1	13.8	88.3	12.1	95.5	13.5	38.2	129.3
BLIP [114]	129M	114.9	15.2	112.1	14.9	115.3	14.4	113.2	14.8	40.4	136.7
OFA [206]	20M	-	-	-	-	-	-	-	-	43.9	145.3
Flamingo [3]	1.8B	-	-	-	-	-	-	-	-	-	138.1
SimVLM [210]	1.8B	113.7	-	110.9	-	115.2	-	112.2	-	40.6	143.3
$OPT_{2.7B}$ BLIP-2 [115]	4M	115.3	15.0	111.0	14.6	112.5	14.0	111.9	14.5	41.8	140.4
$OPT_{2.7B}$ Ours	4M	<u>118.3</u>	<u>15.3</u>	<u>114.7</u>	<u>14.9</u>	<u>114.1</u>	<u>14.1</u>	<u>115.1</u>	<u>14.8</u>	<u>42.3</u>	<u>141.8</u>
$OPT_{2.7B}$ BLIP-2† [115]	129M	**123.0**	**15.8**	**117.8**	**15.4**	**123.4**	**15.1**	**119.7**	**15.4**	**43.7**	**145.8**

Table 3.2: Comparison with different captioning methods on NoCaps and COCO. All methods optimize the cross-entropy loss during fine-tuning. C: CIDEr, S: SPICE, B: BLEU. †: numbers taken from Li et al. [115].

3.4.2. Fine-tuned image captioning

We further fine-tune our pre-trained model for MSCOCO [122] image captioning, employing the text prompt "a photo of ". Following BLIP-2, we fine-tune the model for 5 epochs using a batch size of 1024 (8×128), AdamW with an initial learning rate of $1e^{-5}$, minimum learning rate of 0, warm-up learning rate of $1e^{-8}$ and 1000 warm-up steps, with linear warm-up and cosine scheduling. We evaluate our fine-tuned model on the Karpathy test split of MSCOCO. Also, zero-shot transfer results on the NoCaps dataset [2] are reported. Shown in Table 3.2, our framework improves BLIP-2 in all metrics, with greater improvements in CIDEr compared to SPICE.

Task	Pre-training objectives	Image → Text R@1	Image → Text R@5	Text → Image R@1	Text → Image R@5
Flickr30K	$\mathcal{L}_{\text{BLIP2-stage1}}$	**94.3**	**99.8**	82.9	95.5
	$\mathcal{L}_{\text{BLIP2-stage1}} + \mathcal{L}_{\text{alignment}}$	93.7	99.7	**83.0**	**95.8**
MSCOCO	$\mathcal{L}_{\text{BLIP2-stage1}}$	78.4	93.8	**60.5**	**83.0**
	$\mathcal{L}_{\text{BLIP2-stage1}} + \mathcal{L}_{\text{alignment}}$	**78.7**	**94.5**	60.4	82.8

Table 3.3: Comparison with different image-to-text and text-to-image retrieval methods.

3.4.3. Zero-shot image-text retrieval

While our proposed method primarily focuses on refining visual prompts for a frozen LLM to generate corresponding text, it may not prove as beneficial for image-text retrieval tasks (the ITC and ITM losses are principally responsible for these tasks). Nevertheless, we present results on zero-shot MSCOCO, and zero-shot Flickr30K [152] image-to-text and text-to-image retrievals. We compare two models trained with $\mathcal{L}_{\text{BLIP2-stage1}}$ (ITC, ITM and ITG) and $\mathcal{L}_{\text{BLIP2-stage1}} + \mathcal{L}_{\text{alignment}}$, without any further task-specific fine-tuning. As expected, Table 3.3 reveals that the newly introduced $\mathcal{L}_{\text{alignment}}$ offers limited benefits for retrieval tasks. However, it does not negatively impact the performance.

3.4.4. Ablation studies

Impact of alignment loss weights We investigate the influence of ω_1 and ω_2 in Equation 3.10 and 3.11. $\omega_1 = 0$ and $\omega_2 = 0$ refers to BLIP-2, and $\omega_1 = 10$ and $\omega_2 = 100$ refers to our default configuration of BLIP-2 + P-Former. The alignment loss introduced by the P-Former proves beneficial in both stages of VL pre-training, as shown in Table 3.4.

Alternate language model In this section, we substitute the decoder-based $\text{OPT}_{\text{2.7B}}$ model with an encoder-decoder-based $\text{FLAN-T5}_{\text{XL}}$ as the new LLM. The experiments are conducted with a limited computational budget on 3 × RTX-A6000

ω_1	ω_2	VQAv2 val	OK-VQA test	GQA test-dev
0	0	46.8	25.9	30.5
10	0	<u>51.4</u>	<u>29.2</u>	32.8
0	100	50.4	28.7	<u>33.0</u>
10	100	**52.6**	**30.0**	**34.0**

Table 3.4: Ablations on ω_1 and ω_2 of Equation 3.10 and 3.11 (using $\text{OPT}_{2.7\text{B}}$ as LLMs).

Models	#Pretrain Image-Text	VQAv2 val	OK-VQA test	GQA test-dev
$\text{Flan-T5}_{\text{XL}}$ BLIP-2‡	4M	48.3	31.5	36.4
$\text{Flan-T5}_{\text{XL}}$ ours‡	4M	<u>54.9</u>	<u>35.7</u>	<u>40.3</u>
$\text{Flan-T5}_{\text{XL}}$ BLIP-2†	129M	**62.6**	**39.4**	**44.4**

Table 3.5: Experiments using $\text{Flan-T5}_{\text{XL}}$ as LLM. ‡: using much less GPUs/epochs compared to Sec.3.4.1. †: from Li et al. [115].

and for 5 epochs on both stage 1 and stage 2. The results, displayed in Table 3.5, verify the effectiveness of our framework with another LLM.

Effect of P-Former's pre-training sentence datasets In our primary experiments, we utilize a dataset containing 12M sentences for P-Former training. We investigate the impact of the pre-training sentence dataset for P-Former by re-training it with 4M sentences from our VL pre-training datasets. We then train BLIP-2 + P-Former and report zero-shot VQA results in Table 3.6. This examination underscores that both the implicit decoupling of BLIP-2's two-stage training into a 3-stage training (pre-training of P-Former), and the employment of additional unimodal sentences contribute to the improved outcomes.

3.4.5. Video captioning

Our framework is modality-agnostic with respect to the visual encoder and vision-to-language adaptor, making it applicable to other modalities, such as video. Consequently, we establish a video learning pipeline, with the vision encoder set as a frozen

P-Former	#Pretrain Sentences	VQAv2 val	OK-VQA test	GQA test-dev
×	-	46.8	25.9	30.5
✓	4M	<u>51.7</u>	<u>28.2</u>	<u>32.3</u>
✓	12M	**52.6**	**30.0**	**34.0**

Table 3.6: Ablations on sentence datasets used to train P-Former (using $\text{OPT}_{2.7\text{B}}$ as LLMs). The first row w/o P-Former is baseline BLIP-2.

	BLEU-4	CIDEr	ROUGE
NITS-VC [181]	20.0	24.0	42.0
ORG-TRL [244]	32.1	49.7	48.9
$\mathcal{L}_{\text{ITG}}$	29.3	56.6	48.2
$\mathcal{L}_{\text{ITG}} + \mathcal{L}_{\text{alignment}}$	**30.9**	**60.9**	**49.1**

Table 3.7: VATEX English video captioning. Baseline is a sequential model (I3D → Transformer → $\text{OPT}_{2.7\text{B}}$), training end-to-end with ITG.

I3D [18] video encoder, the vision-to-language adaptor as a Transformer-base, and the LLM decoder as the $\text{OPT}_{2.7\text{B}}$ (also frozen). We then train this model on the VATEX [208] English training set and evaluate it on the validation set. This dataset contains 26K videos for training. The experiments are conducted on an RTX-A6000. Initially, we train the model solely using $\mathcal{L}_{\text{alignment}}$ for 10 epochs with the P-Former, followed by end-to-end learning with $\mathcal{L}_{\text{ITG}}$ for an additional 10 epochs.

Our baseline, represented in Table 3.7, is competitive with two well-established video captioning models: MITS-VC [181] and ORG-TRL [244]. It is noteworthy that the current state-of-the-art on this benchmark, VideoCoCa [221], is trained on 10M videos, in contrast to our model, which is trained on merely 26K videos. Furthermore, the integration of P-Former and $\mathcal{L}_{\text{alignment}}$ enhances the CIDEr score by 4.3 (from 56.6 → 60.9).

Despite being a smaller-scale experiment without large-scale pre-training, we demonstrate that our learning framework can be generalized to another modality (i.e., video-learning), employing a different vision-language adaptor (i.e., a plain Transformer

as opposed to a Q-Former).

Section 3.5

Summary

This work introduces a novel optimization framework for enhancing vision-language models based on large, frozen LLMs. We observe that the end-to-end image-to-text pre-training can be backwardly decoupled: initially determining the "ideal prompt" that triggers the LLM to generate the target text (which can be trained in an unsupervised fashion), followed by the alignment of visual features to the prompt. To this end, we train a P-Former, which functions similarly to a semantic sentence embedding model, to predict prompts to which visual features should align. Experimental results demonstrate that including alignment loss (via P-Former) in the BLIP-2's framework significantly narrows the performance gap between models trained with 4M and 129M image-text pairs.

Section 3.6

Supplementary Information

Intuition and motivation behind P-Former In this section, we summarize the intuitive explanation and motivation on why learning an ideal language prompt helps more than using visual ones as in the counterpart models.

- In our experiments with base models like BLIP-2, the architecture consists of three sequential components: (1) ViT, (2) VL-connector, and (3) LLM decoder. Since we use a frozen LLM for generation, optimizing closer to the LLM decoder becomes more pivotal for achieving optimal generation quality.
- The unique design of P-Former mirrors a sentence embedding model. This means the prompts predicted by the P-Former carry rich semantics. Therefore, during

evaluations on unfamiliar images, the model boasts an improved generalization capability.

- BLIP2's studies indicate that direct end-to-end optimization of the sequential model can sometimes lead to catastrophic forgetting. Our approach adds an additional layer of complexity by decomposing the 2-stage BLIP2 training into 3 stages, further addressing this optimization challenge.
- For BLIP2, optimization of soft prompt is learned only using text from image-text pair, while our decoupled training allows for leveraging additional unimodal data for optimizing these soft prompts

Justification for lack of ablation experiments w/ and w/o the P-Former We purposely omitted experiments with and without the P-Former module (e.g., using a randomly initialized prompt p). This omission was driven by the following considerations:

- **Random initialization and learning without P-Former**: Our initial approach was to directly learn from a randomly initialized prompt p without incorporating the P-Former. But, upon testing, we identified a significant challenge. For a smaller model variant like opt-2.7b, which possesses a hidden size of 2560, if we employ 32 tokens as soft prompts for an expansive dataset with 4M sentences, the resultant model would have to accommodate an overwhelming 327B parameters. This would have computational implications and potentially overfit, as learning from such a vast parameter space can dilute the essential semantic connections between various sentences.
- **P-Former's efficiency in parameterization**: The P-Former emerged as a solution to this parameter explosion problem. Instead of requiring a unique prompt for each data point in the dataset, the P-Former parameterizes the soft prompt p using a semantically-rich Transformer model. This design ensures that the total

number of parameters remains fixed at 110M. The major advantage here is scalability. Whether working with a dataset of 4M, 12M, or even larger (e.g., 129M) or LMs with varying decoder sizes, the P-Former guarantees a consistent number of parameters, making the model more computationally efficient and preventing the loss of essential semantic relationships.

In brief, our experimentation strategy was driven by the dual goals of maintaining computational efficiency while preserving rich semantics. The challenges posed by direct learning from a randomly initialized prompt emphasized the need for a more structured approach, leading to the birth of the P-Former concept.

Qualitative analysis on VQA In this section, we incorporate qualitative comparisons for the GQA and OKVQA datasets, allowing us to offer more nuanced insights. In Figure 3.4, we show several examples comparing our model's response with BLIP-2 and the ground truth (GT). From these examples, it can be observed that there is greater agreement with GT by our model.

It should be noted that the abstract semantic reasoning of our model can sometimes lead to artificially low scores for our model when looking for an exact match. For instance, asking "What occupation might he have?" with a picture of a person driving a forklift generates the answer "forklift operator" by our model, whereas the correct exact answer in the GT is stated as "forklift driver." Though these two answers are semantically identical, they will count as a wrong generation by our model.

Additional discussion of the results In this section, we provide more interpretation of the results. For instance, Table 3.1, in addition to underscoring the potency of our proposed framework in bolstering the zero-shot VQA performance, particularly when trained with 4M image-text pairs, shows that our method manages to considerably close the performance gap between the BLIP-2 trained on different scales: 4M

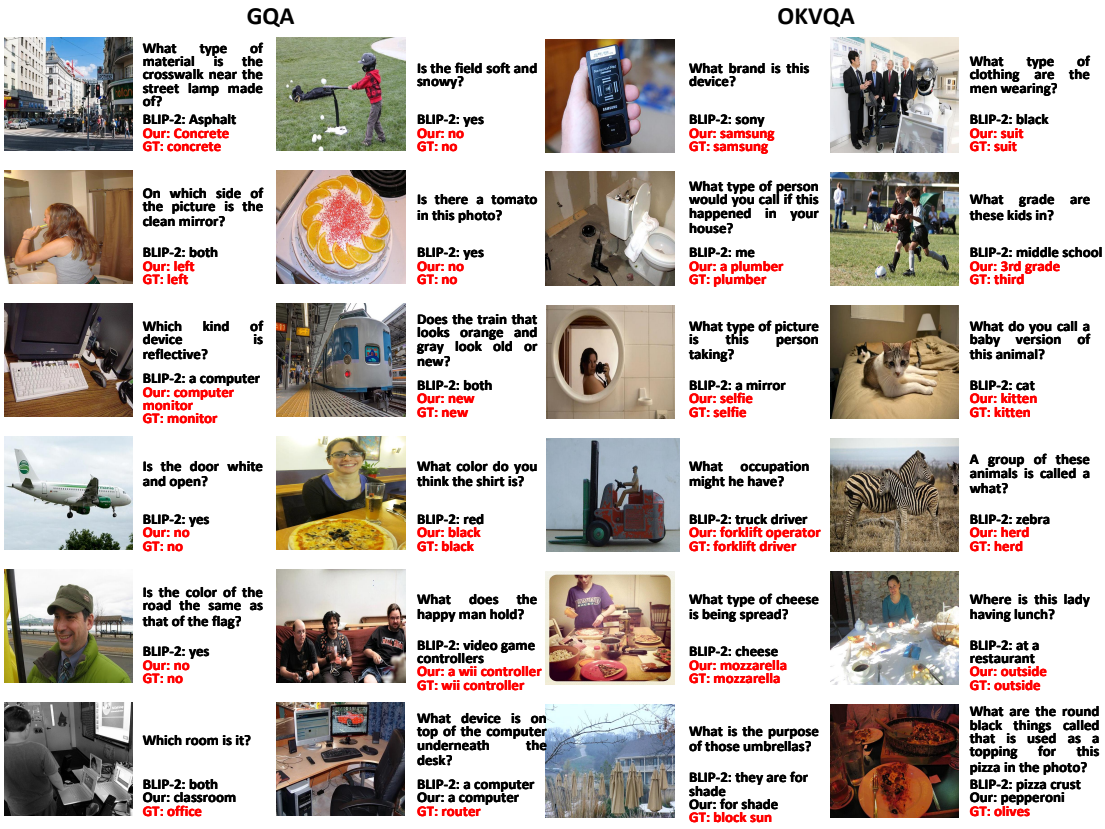

Figure 3.4: Qualitative analysis on success and failure cases of GQA and OKVQA.

and 129M image-text pairs. This suggests that the effectiveness of our model is not solely a function of the amount of training data but rather the methodology itself. In essence, this table illustrates how strategic modifications and improvements can achieve comparable results to models trained on much larger datasets.

Similarly, Table 3.2 provides insights into our model's adaptability. When we fine-tune our pre-trained model for a specific task like MSCOCO image captioning, the results reflect an overall enhancement over BLIP-2 across all metrics. The pronounced improvement in CIDEr, as opposed to SPICE, indicates that our model is adept at recognizing and generating more relevant and contextually accurate descriptions of images. The additional data on zero-shot transfer to the NoCaps dataset further substantiates the model's capability to generalize and adapt to newer, unseen data.

Finally, while our model's primary design goal is to refine visual prompts for text generation, Table 3.3 offers a perspective on its performance in the retrieval domain. Even though the model was not specifically optimized for retrieval tasks, it is evident that the introduced modifications do not compromise the retrieval performance, attesting to the model's robustness.

LLM-dependence of the stage-1 pre-training It should be noted that our stage-1 pre-training needs to be repeated for each LLM, if $\omega_1 \neq 0$. However, as evidenced in Table 3.4 ($\omega_1 = 0$ and $\omega_2 = 100$), our approach achieves competitive results even without the alignment loss in stage-1, focusing the alignment solely on stage-2.

Chapter 4

VLM Pre-training with Less Compute

In this Chapter, we present SimVLG (SimVLG: Simple and Efficient Pretraining of Visual Language Generative Models [86]). This work has been published in arXiv pre-print server.

Section 4.1

Overview

In this chapter, we propose "SimVLG", a streamlined framework for the pre-training of computationally intensive vision-language generative models, leveraging frozen pre-trained large language models (LLMs). The prevailing paradigm in vision-language pre-training (VLP) typically involves a two-stage optimization process: an initial resource-intensive phase dedicated to general-purpose vision-language representation learning, aimed at extracting and consolidating pertinent visual features, followed by a subsequent phase focusing on end-to-end alignment between visual and linguistic modalities. Our one-stage, single-loss framework circumvents the aforementioned computationally demanding first stage of training by gradually merging similar visual tokens during training. This gradual merging process effectively compacts the visual information while preserving the richness of semantic content, leading to fast conver-

gence without sacrificing performance. Our experiments show that our approach can speed up the training of vision-language models by a factor $\times 5$ without noticeable impact on the overall performance. Additionally, we show that our models can achieve comparable performance to current vision-language models with only 1/10 of the data.

Section 4.2

Motivation

The landscape of vision-language modeling has undergone significant transformations in recent years, with CLIP [154] serving as a landmark development. It distinguished itself through unparalleled zero-shot classification capabilities and efficiency in image-text retrieval tasks. Successive models like ALBEF [113], X-VLM [231], and VLMo [9] further broadened the scope, addressing a myriad of tasks such as retrieval, visual entailment, and closed-set Visual Question Answering (VQA), among others.

Recently, the field has been enriched by the advent of generative models designed for complex image-to-language tasks. Notable contributions include CoCa [228], SimVLM [210], Frozen [199], Flamingo [3] and BLIP-2 [115], targeting tasks like image and video captioning and open-set VQA. Specifically, CoCa demonstrates robust performance across both uni-modal and multi-modal tasks, leveraging a large-scale dataset for training from scratch.

The computationally intensive nature of pre-training Vision-Language Models (VLMs) led to the conceptualization of BLIP-2. This model seeks to alleviate computational costs by employing pre-trained vision encoders (ViT) and language decoders (LLM). As illustrated in Figure 4.1a, a central innovation in BLIP-2 is the *Q-former*, a vision-language connector outfitted with learnable queries for effective cross-attention mechanisms. This architectural choice, however, demands an intensive pre-training regimen, referred to as *BLIP-2's Stage 1*. The stage involves

three learning objectives—image-text contrastive, image-text matching, and language generation—and necessitates multiple forward passes for optimization.

Despite its efficiency gains over CoCa, BLIP-2's training still imposes considerable computational costs. This poses challenges for research environments with limited computational resources, such as university labs. Our experiments indicate that the Stage-1 training of BLIP-2 took approximately eight days on eight A100-80G GPUs. This computational burden has consequently restricted research to using the pre-trained Q-former, hindering the exploration of alternative ViTs in VLMs. This limitation is evident in subsequent works like InstructBLIP [30], VideoChat [116], Video-LLaMA [236], X-LLM [21].

The prospect of reducing BLIP-2's computational cost through end-to-end, single-stage training is compelling. Such an approach would remove the complexities associated with resource allocation and hyper-parameter tuning inherent in multi-stage training. Yet, direct end-to-end training with BLIP-2 poses substantial challenges, corroborated by both original findings from BLIP-2 and our own empirical analyses. We hypothesize that these challenges emanate from the intrinsic design of the Q-former. Specifically, the inclusion of randomly initialized learnable queries and cross-attention mechanisms complicates the optimization landscape, especially when the aim is to minimize the representational disparity between visual and linguistic modalities.

In this chapter, we propose an alternative to the Q-former, employing a systematic token-merging [13] strategy that is both intuitive and effective. Here, token merging signifies the step-wise aggregation of tokens with analogous features across the layers of the Transformer model (see Figure 4.1c). We substitute the Q-former in BLIP-2 with a standard Transformer architecture augmented with token merging capabilities, which we term *TomeFormer*. Importantly, the TomeFormer is trainable to function effectively as an efficient vision-language connector. This modification, which we

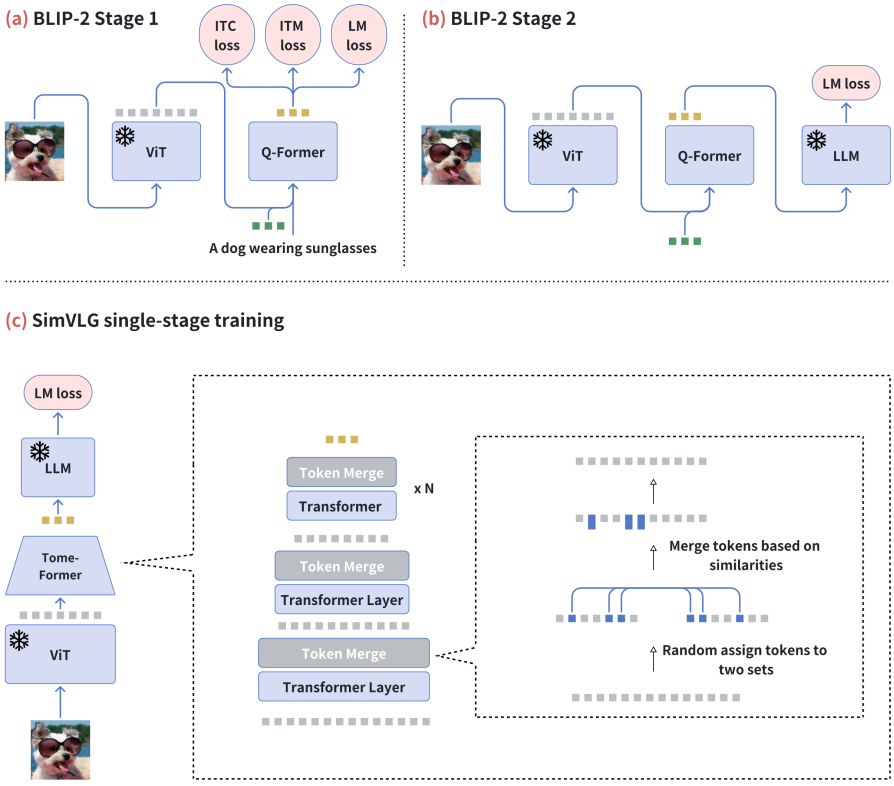

Figure 4.1: Overview of the BLIP-2 and SimVLG. **(a)** The 1st stage of BLIP-2 training involves a complex and computationally intensive process. It employs learnable queries (depicted in green) to select 32 tokens (in orange) from an extensive pool of 256 visual features (shown in grey). This queried output is then utilized to fulfill three distinct learning objectives, necessitating multiple forward passes within a single optimization step. **(b)** The 2nd stage of BLIP-2 adopts a conventional end-to-end training approach, mapping images directly to text. **(c)** In contrast, SimVLG employs a streamlined, single-stage training mechanism with a unified loss. Here, visual tokens (in grey) are progressively aggregated based on their inherent similarities at each layer of the TomeFormer architecture. The final set of merged tokens (in orange) serves as semantically rich but computationally efficient soft prompts, guiding the LLM to generate a corresponding caption for the input image.

call **Sim**ple **V**isual **L**anguage **G**enerative pre-training model (SimVLG), facilitates a streamlined, single-stage training process. It requires only a singular learning objective and a single forward pass per optimization step. This stands in contrast to BLIP-2's multi-stage training, laden with multiple objectives and several forward passes.

Further, we introduce a *soft attentive temporal* token fusion mechanism within the ViT for effective video-language modeling. This eliminates the need for modality

realignment, contrasting approaches such as the temporal Q-former [236], or the addition of new learnable temporal queries [116]. Our strategy simplifies the optimization challenges tied to working with relatively smaller video-text datasets, compared to their image-text counterparts. Remarkably, we demonstrate that even without video pre-training, our temporal token fusion approach can effectively train robust video-language models. This differs from recent work in video-language models that depend on pre-training models using vast million-scale video-text datasets.

Our contributions are summarized as follows:

- We adapt token merging, initially designed to enhance ViT inference speed without training, to serve as a means for condensing semantically-rich visual features within the vision language connector. Concurrently, we present a novel temporal token merging scheme for video modeling.
- Our proposed image-text model featuring TomeFormer competes effectively with BLIP-2, while requiring just a fraction of the computational resources. Given the reliance on BLIP-2's pre-trained model in contemporary studies, our approach widens the exploratory scope for various ViTs.
- We introduce a straightforward spatial attentive temporal modeling technique that allows for the seamless adaptation of pre-trained image-text models to video-text tasks. This approach eliminates the need for complex modality re-alignment, a common requirement in alternative methods.

Section 4.3

Technical Approach

We begin by presenting our image-text model and then describe the adaptations made to this pre-trained model for video-related tasks.

Three key models serve as the foundation for generative tasks in the image-text

domain: CoCa, Flamingo, and BLIP-2. The high computational cost of training CoCa from scratch and the proprietary nature of Flamingo led us to adopt the BLIP-2 framework. BLIP-2 utilizes frozen pre-trained ViT and LLM for vision-language tasks. Notably, BLIP-2 has gained significant traction in the field due to its open-source availability, thereby influencing subsequent research in various projects such as InstructBLIP, VideoChat, Video-LLaMA, X-LLM, Valley, and Video-ChatGPT.

BLIP-2 Framework In BLIP-2, a ViT serves as the vision encoder, ingesting images and outputting a set of 257 visual tokens. A specialized component, the Q-former, is then added to the ViT. This Q-former utilizes 32 learnable queries to selectively extract and transform the 32 most informative tokens from the ViT's output pool of 257 tokens. These 32 tokens are subsequently used as soft language prompts to guide the LLM in generating text that describes the image content.

Although it is theoretically possible to train BLIP-2 end-to-end, empirical evidence suggests that such an approach often yields suboptimal outcomes. Consequently, BLIP-2 employs a preliminary Stage-1 pre-training phase for both the ViT and Q-former. During this phase, three learning objectives—Image-Text Contrastive (ITC) loss, Image-Text Matching (ITM) loss, and Language Modeling (LM) loss—are optimized simultaneously. This Stage-1 training involves 250,000 steps, each requiring multiple forward passes due to the multiple learning objectives.

Despite being more efficient than CoCa, our observations reveal that BLIP-2's Stage-1 training still demands approximately eight days on a server equipped with eight A100-80G GPUs. This computational cost poses a constraint for subsequent research that aims to explore various ViT configurations, as each new configuration requires a complete restart of the Stage-1 training process.

We introduce SimVLG-Image (abbreviated as SimVLG, and shown in Figure 4.1c), an optimized vision-language generative pre-training model. Similar to BLIP-2,

SimVLG utilizes a ViT for visual encoding and an LLM for linguistic decoding. The key innovation is the incorporation of a standard Transformer, augmented with spatial Token Merging, to act as the connector between the visual and linguistic modalities.

Formally, our framework includes a vision encoder E_{vision}, which ingests an input image I and encodes it into a fixed set of visual tokens: $[v_1, v_2, ...v_L] = E_{\text{vision}}(I)$. Here, L denotes the number of image patches. Subsequently, we employ a Transformer equipped with token-merging modules, termed as *TomeFormer* ($T_{\text{v}\to\text{t}}$) as the vision-to-language connector. This module effectively compresses the token count: $[v'_1, v'_2, ...v'_{L'}] = T_{\text{v}\to\text{t}}(f_{\text{proj}_1}([v_1, v_2, ...v_L]))$. In this equation, L' is considerably smaller than the initial token count L[1]. The LLM decoder then employs these compressed tokens as soft prompts for text generation: output $= D_{\text{LLM}}(f_{\text{proj}_2}([v'_1, v'_2, ...v'_{L'}]))$. Projection functions f_{proj_1} and f_{proj_2} are used to ensure dimension compatibility. Three main advantages of using TomeFormer are:

- Efficient token reduction, facilitating the transformation of loosely-structured visual data into a more concise yet informative representation.
- Computational efficiency, as the uncompressed ViT output consists of 256 tokens, plus a [CLS] token. Without compression, the subsequent vision-to-language connector would be computationally expensive in terms of both memory and processing power.
- Semantic richness of the compressed tokens. Unlike BLIP-2, which requires an extensive pre-training phase for feature extraction via its Q-former, TomeFormer naturally merges semantically similar tokens. Our empirical evidence confirms that TomeFormer-equipped models train more efficiently compared to alternatives like BLIP-2.

[1]We merge 19 tokens at each layer of the TomeFormer. Thus, 256 visual tokens are reduced to 28 tokens. Ablation on the number of merged tokens at each layer is studied in Section 4.5

Section 4.4

Experiments

Our experimental setup is as follows:

- **Pre-training Data** Our model is pre-trained using the MSCOCO [122] and CapFilt [114] datasets, which include BLIP's pseudo-labeled Conceptual Captioning [173], SBU [142], and LAION [170] datasets—similar to the data sources utilized in BLIP-2. Note that we intentionally exclude the VG [101] dataset from our pre-training procedure, as it mainly consists of localized captions.
- **Models** In order to facilitate a direct and fair comparison with BLIP-2, we employ the same ViT, denoted as `eva-vit-g` [47]. For the language model decoders, we explore both `opt-2.7b` [242] and `vicuna-7b` [26]. Our TomeFormer is initialized using `bert-base-uncased`, ensuring parameter count parity with BLIP-2's Q-former. The only difference in parameterization between our model and BLIP-2 lies in the additional 32 learnable queries present in the latter.
- **Pre-training Details** Our pre-training setup closely mirrors the configurations of BLIP-2. We utilize a maximum learning rate of 1×10^{-4} and a minimum learning rate of 1×10^{-5}. The learning rate follows a schedule that begins with a linear warm-up phase of 5000 steps starting from 1×10^{-6} and then transitions to a cosine decay schedule. Weight decay is set at 0.05. The training is conducted with a batch size of 1600, distributed over either 8× A100-80G or 32× V100-32G.
- **Downstream Tasks** SimVLG-Image is evaluated without additional fine-tuning on a variety of tasks, including MSCOCO captioning, VQAv2 [64], GQA [75], and OK-VQA [134]. For video tasks, SimVLG-Video is evaluated on fine-tuned MSR-VTT [216] and MSVD [20] captioning tasks.

We conducted comparative evaluations between SimVLG and BLIP-2 on multiple

image-text benchmarks, including zero-shot VQAv2, GQA, OK-VQA, and MSCOCO captioning. It is essential to note that BLIP-2 demands an extensive Stage-1 pre-training phase involving 250,000 optimization steps. This phase incorporates three distinct loss functions and necessitates multiple forward passes through the model, a process crucial for BLIP-2 to prevent model divergence.

Table 4.1 summarizes the results of our experiments. Our primary insights can be distilled into the following key points:

- Utilizing the same training set of 104 million image-text pairs and an equal number of optimization steps (250,000), SimVLG consistently outperforms BLIP-2 across nearly all evaluated tasks.
- Remarkably, SimVLG maintains competitive performance even when its training budget is trimmed to approximately one-third of BLIP-2's, specifically 150,000 optimization steps.
- Our experiments show that SimVLG can produce satisfactory results with a significantly reduced training dataset of 11 million image-text pairs, while still undergoing 150,000 optimization steps.
- SimVLG retains its efficacy even when the training budget is restricted to as few as 90,000 steps, demonstrating the model's efficiency and robustness.

Training Time In the Stage-1 pre-training phase, BLIP-2 requires considerable time, necessitating multiple forward passes to optimize three separate loss functions. We document the training durations for both BLIP-2 and SimVLG when utilizing eight A100-80G GPUs in Table 4.2.

Although BLIP-2 significantly reduces training time relative to predecessors like CoCa, it still mandates an extended training duration—approximately ten days. This extensive time commitment limits the feasibility of researchers to investigate various ViT configurations. Most subsequent works based on BLIP-2 continue to use the

Table 4.1: Comparison of methods on zero-shot VQA and MSCOCO captioning tasks without additional fine-tuning. †: We were able to download approximately 81% of LAION-115M and 78% of CCS-14M from the CapFilt dataset. ‡: BLIP-2 incorporates an additional set of 32 learnable queries, each with a dimension of 768.

	Models	# pre-train image-text	# trainable params	# training steps	VQAv2 val	GQA test-dev	OK-VQA test	MSCOCO val
OPT-2.7b	BLIP-2	104M†	110M+‡	250k + 80k	44.6	30.6	26.0	137.7
	SimVLG	104M	110M	250k	**48.4**	**30.9**	**27.2**	**139.1**
	SimVLG	104M	110M	150k	46.9	30.8	24.8	137.0
	SimVLG	11M	110M	150k	46.3	30.0	23.0	135.1
	SimVLG	104M	55M	90k	45.9	30.6	25.8	134.0
Vicuna-7b	BLIP-2	104M†	110M+‡	250k + 80k	**57.8**	35.7	27.8	138.0
	SimVLG	104M	110M	250k	54.8	35.6	30.4	**139.1**
	SimVLG	104M	110M	150k	55.5	**36.3**	**30.6**	137.9
	SimVLG	11M	110M	150k	54.6	34.0	27.3	138.0
	SimVLG	104M	55M	90k	53.4	34.7	30.6	137.8

Table 4.2: Runtime comparison of BLIP-2 and SimVLG when utilizing OPT-2.7b as the LLM.

Models	Stage 1 (5k steps)	Stage 2 (5k steps)	# steps	Clock time	MSCOCO
BLIP-2	3 hrs 50 min	2 hrs 40 min	330k	234 hrs	137.7
SimVLG	-	2 hrs 45 min	250k	133 hrs	139.1
SimVLG	-	2 hrs 45 min	150k	80 hrs	137.0
SimVLG (55M)	-	2 hrs 35 min	90k	47 hrs	136.8

pre-trained Q-former in conjunction with the `eva-vit-g` model, thereby narrowing the scope of ViT exploration. In contrast, SimVLG significantly trims the training time while maintaining satisfactory performance levels, thus providing researchers with the latitude to explore a wider array of advanced ViTs in future investigations.

Ablation on Visual Encoders One of the limitations of BLIP-2 is that it requires an extensive stage-1 pre-training for every different vision encoder. This prohibits practitioners from exploring stronger ViTs when they are available. SimVLG offers fast training of models, allowing for exploration of different ViTs as visual encoders.

We conduct an ablation experiment on two ViTs (CLIP_L and EVA-ViT_G) using 8× RTX-A6000 and the CCS-14M dataset for pre-training. The models are trained for 60,000 steps.

Shown in Table 4.3, SimVLG is robust to different visual encoders, and the stronger

LLM	ViT	VQA	GQA	OK	COCO
OPT	CLIP_L	44.7	**30.9**	22.7	123.9
	EVA-ViT_G	**45.2**	30.6	**22.8**	**130.6**
Vicuna	CLIP_L	49.0	33.0	23.6	125.2
	EVA-ViT_G	**52.5**	**34.6**	**27.9**	**132.4**

Table 4.3: Ablation studies on different visual encoders of SimVLG. VQA→VQAv2, OK→OKVQA, COCO→MSCOCO (CIDEr).

ViT leads to better results. This implies that while SimVLG also requires retraining for different ViTs, but the single-stage training and quick convergence allow it to benefit from a future release of the latest ViTs, given its capability of fast adaptation.

Section 4.5

Analysis

Impact of Soft Prompt Length Within the TomeFormer, the vision-to-language connector in SimVLG, we introduce a hyperparameter r that regulates the number of spatial tokens merged at each layer. Increasing r substantially reduces the token count, but runs the risk of eliminating important visual details. On the other hand, a smaller r produces two main effects: (1) a more diffuse representation of visual features, complicating the optimization landscape, and (2) elongated soft prompts for the LLM, leading to increased computational cost during training, such as memory overflow and extended training durations.

To study the effects of r, we conduct an ablation experiment using 8× RTX-A6000 and the CCS-14M dataset for pre-training. The models are trained for 60,000 steps, and their performance is evaluated using CIDEr scores on MSCOCO captioning. In Figure 4.2, we observe that a smaller r (e.g., 10) places a higher computational load on both TomeFormer and the LLM, extending training time and compromising optimization, as evidenced by lower CIDEr scores. In contrast, a larger r value (e.g.,

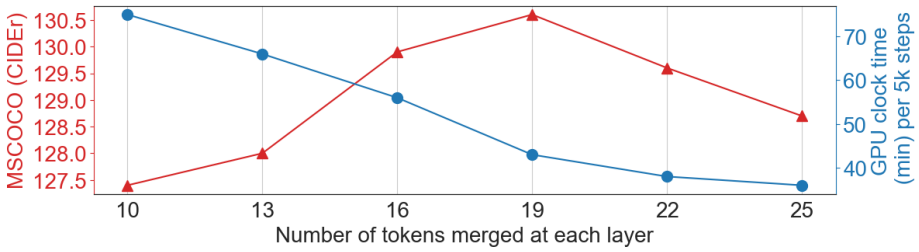

Figure 4.2: Trade-off between MSCOCO captioning scores (depicted in red) and GPU training time (depicted in blue) as a function of the number of tokens merged (r) in TomeFormer. A larger r value leads to shorter soft prompts in the LLM, thereby decreasing computational time (blue line). However, overly compressed soft prompts may result in the loss of valuable visual information, while insufficiently compressed features complicate the optimization process.

25) expedites training but at the expense of model performance, likely due to excessive feature compression and consequent information loss.

Token Merging Visualization in SimVLG One notable advantage of SimVLG over BLIP-2 is the absence of a requisite Stage-1 pre-training for the vision-to-language connector. This simplifies the training pipeline by removing the need to train the model to extract text-informative visual features. We posit that the token merging process in TomeFormer naturally aggregates tokens associated with visually similar elements, thereby yielding concise yet semantically rich visual features from the onset of training. This inherent capability allows SimVLG to benefit from a more streamlined, single-stage training regimen with just one learning objective.

Essentially, our token merging strategy serves as an efficient approximation of QFormer's functionality, compressing visual features in a semantically meaningful manner. Figure 4.3 illustrates this, displaying the visual tokens before and after training with our TomeFormer. The figure shows that the compressed visual features obtained via token merging are semantically informative and offer basic object segmentation within the image. Furthermore, the semantic coherence of these merged tokens improves as training advances.

Figure 4.3: Pre- and post-training visualization of merged tokens in SimVLG. The visual features compressed via token merging exhibit semantic informativeness even prior to training. This inherent characteristic facilitates SimVLG's ability to converge quickly in an end-to-end training setup.

Section 4.6

Summary

This chapter introduces SimVLG, an efficient and streamlined pre-training framework for vision-language generative models. Like BLIP-2, SimVLG employs frozen ViT and LLM. It further leverages a conventional Transformer architecture with token-merging capabilities, known as TomeFormer, to act as the vision-to-language connector. Compared to BLIP-2, SimVLG offers the distinct advantage of one-stage training. This reduces computational overhead and maintains competitive performance even with only 1/3 to 1/6 of the computational budget required by BLIP-2.

SimVLG demonstrates the possibility of achieving state-of-the-art performance in vision-language tasks without the need for complex training regimens or high computational budgets. This work thus makes a significant contribution to the ongoing efforts to develop more accessible, efficient, and powerful models for understanding and generating visual and textual information.

Section 4.7

Supplementary Information

Technical Details of Token Merging In this section, we briefly summarize the technical designs of Token Merging (ToMe) [13]. Token Merging was initially proposed in Bolya et al. [13] for accelerating ViTs without training. Whereas we re-purpose ToMe to condense the visual features used as language prompts in the LLM. Please refer to Sections 3 of Bolya et al. [13] for full details.

Strategy. The token merging operations take place in between the attention and MLP blocks of each Transformer layer. ToMe reduces r tokens per layer. And over the L layers of a Transformer, it reduces a total of $r \times L$ tokens. In our experiments, we set $r = 19$ and our TomeFormer has 12 layers.

Token Similarity. The similarities of tokens are defined by the cosine similarity (dot product) of keys of tokens.

Bipartite Matching. The bipartite soft matching algorithm is summarized as follows:

- Tokens are randomly partitioned into two sets $\mathbb{A}$ and $\mathbb{B}$.
- Each token in set $\mathbb{A}$ is linked to the most similar token in set $\mathbb{B}$.
- Keep links with top r similarities.
- Merge tokens with top r links.
- Concatenate set $\mathbb{A}$ and $\mathbb{B}$ back into a single set.

Details on Our Implementations of BLIP-2 and VideoCoCa Our reported results of our re-trained BLIP-2 are slightly worse than what was reported in Li et al. [115]. There are mainly three reasons:

r	VQA	GQA	OK	**COCO**
10	45.7	31.3	23.6	127.5
13	46.2	31.4	24.5	128.0
16	46.3	30.9	24.3	129.9
19	45.2	30.7	22.8	130.6
22	45.5	31.5	21.8	129.7
25	44.7	31.1	21.5	128.7

Table 4.4: Ablation studies on r in TomeFormer.

Models	Stage 1 (MACs)	Stage 1 steps	Stage 2 (MACs)	Stage 2 steps
BLIP-2	36.7G	250k	6.28G	80k
SimVLG	-	-	11.9G	250k
SimVLG	-	-	11.9G	150k
$\text{SimVLG}_{55\text{M}}$	-	-	5.6G	90k

Table 4.5: **Multiply–accumulate operations** (MACs) comparison of Q-Former (of BLIP-2) and TomeFormer (of SimVLG) when utilizing OPT-2.7b as the LLM.

- We are only able to download 104M image-text pairs from the original 129M CapFlit dataset.
- We intentionally exclude the VG dataset from our pre-training procedure, as it mainly consists of localized captions. Thus, our re-trained BLIP-2 is more challenging when evaluated on GQA, which is built on VG dataset.
- The exact dataset weighting is unknown from the LAVIS project, we use a weighting that is based on the size of each pre-training dataset, i.e., CSS14M, LAION115M, MSCOCO.

For video captioning in Table 5.1 and Table 5.2, because VideoCoCa is not open-sourced, we use a pre-trained model `OpenCoCa` released by mlfoundations.

Ablations on TomeFormer In this section, we provide experimental results in VQAv2, GQA, and OKVQA of SimVLG, by varying hyper-parameter r in TomeFormer. As we can see from Table 4.4, SimVLG is robust to the choice of r.

Models	Stage 1 time /5k	Stage 2 time /5k	Clock time
BLIP-2	3 hrs 50 min	2 hrs 40 min	234 hrs
SimVLG	-	2 hrs 45 min	133 hrs
SimVLG	-	2 hrs 45 min	80 hrs
$\text{SimVLG}_{55\text{M}}$	-	2 hrs 35 min	47 hrs

Table 4.6: Training time comparison of BLIP-2 and SimVLG when utilizing OPT-2.7b as the LLM.

MACs (FLOPs) in Q-Former and TomeFormer In this section, we compute **multiply–accumulate operations** (MACs) in Q-Former and TomeFormer. MACs performs $a \leftarrow a + (b \times c)$. Whereas, FLOPs is **floating operations** which includes $\times$ / $+$ / $\div$ ... etc. One MACs has one $\times$ and one $+$. And thus, roughly speaking, FLOPs is two times as MACs.

In our experiments, BLIP-2 and SimVLG have identical ViTs and LM decoders. Thus, we only compare the MACs in VL Connector in BLIP-2 and SimVLG (i.e., Q-Former and TomeFormer).

There's a large MACs in BLIP-2 stage-1 due to three forward passes using Q-Former, where the last forward-pass used for caption loss dominates (27.0G). In contrast, SimVLG does not require such a representation training stage (stage-1) at all.

Another reason why BLIP-2 stage-1 is slow is that the computation of Image-Text Contrasive and Image-Text Matching losses needs `concat_all_gather` operations that require GPU communications. Further Image-Text Matching requires binomial sampling of hard negatives. In comparison, our SimVLG circumvents such computations/communications.

Details on Training Time In Table 4.6, we provide training times for different model configurations. For BLIP-2, each training iteration in Stage-1 takes longer

than Stage-2 due to the three forward-passes to compute the Image-Text Contrast, Image-Text Match, and Language Modeling learning objectives.

Chapter 5

Effective Adaptation of Pre-trained Models

Section 5.1

Effective Adaptation of VLMs

In this Section, we present SimVLG-video from SimVLG paper (SimVLG: Simple and Efficient Pretraining of Visual Language Generative Models [86]). This work has been published in arXiv pre-print server. In this work, we delve into the effective adaptation of VLMs pre-trained on extensive image-text datasets for tasks related to video language comprehension. Given the inherent challenge of acquiring high-quality video-text pairs, our investigation bears substantial implications for enhancing the learning process of video-language models.

5.1.1. Overview and Motivation

we introduce a *soft attentive temporal* token fusion mechanism within the ViT for effective video-language modeling. This eliminates the need for modality realignment, contrasting approaches such as the temporal Q-former [236], or the addition of new

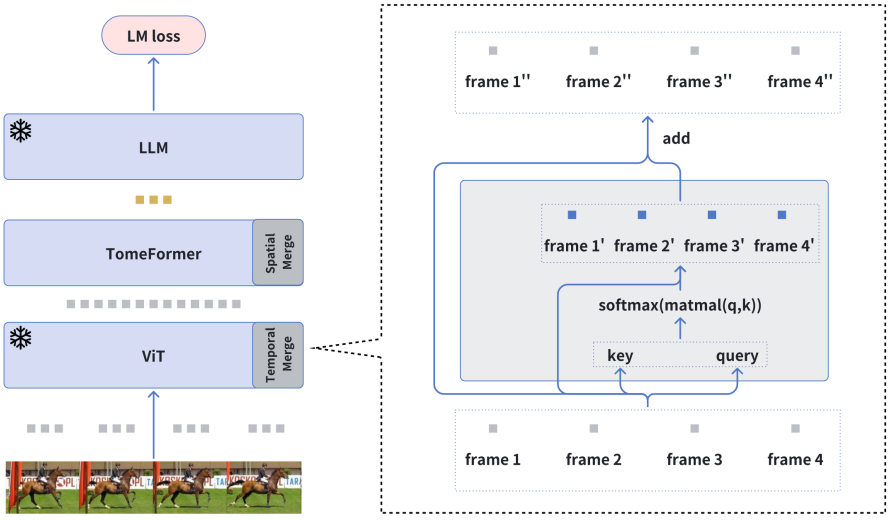

Figure 5.1: Overview of SimVLG-Video: In addition to TomeFormer's spatial token merging capabilities, our design introduces Temporal Attentive Soft Token Merging for nuanced temporal modeling. Each frame's output is calculated as a learnable weighted average of other frames in the video. This approach maintains the integrity of pre-existing, well-trained image-text models. For instance, when the input consists of static videos with identical frames, SimVLG-Video operates as if it were an image-text model. Importantly, this architecture avoids the need for complex model realignment, a requirement in alternative designs that insert a temporal Q-former between the visual encoder and the language model.

learnable temporal queries [116]. Our strategy simplifies the optimization challenges tied to working with relatively smaller video-text datasets, compared to their image-text counterparts. Remarkably, we demonstrate that even without video pre-training, our temporal token fusion approach can effectively train robust video-language models. This differs from recent work in video-language models that depend on pre-training models using vast million-scale video-text datasets.

Although many image-text models can be adapted for video-text tasks with minor modifications, they often overlook the importance of temporal modeling. For example, VideoCoCa extends CoCa using an attentive pooler without changing CoCa's architecture, while InstructBLIP and BLIP-2 employ a concatenated soft-prompt approach. This simplicity comes at the cost of inadequate temporal modeling, which is later addressed by VideoChat and Video-LLaMA through the introduction of

learnable temporal queries and a temporal Q-former. However, these additions disrupt the integrity of the aligned VLM and necessitate a re-alignment process requiring substantial video-text pairs, as shown in VideoChat and Video-LLaMA.

5.1.2. Methods and Experiments

In this chapter, we propose a novel module called *Temporal Attentive Soft Token Merging* to enhance the ViT backbone with temporal modeling capabilities. Formally, let v be a video feature tensor with dimensions $[B \times N \times L \times D]$, where B is the batch size, N is the number of frames, L is the sequence length (i.e., the number of patches in a single video frame), and D is the hidden dimension. Initially, we reshape v into $[(B \times N) \times L \times D]$ which is subsequently fed into the self-attention layer of the ViT for *spatial modeling* as:

$$v' = \text{self-attn}(v.\text{reshape}(B \times N, L, D)) \tag{5.1}$$

For *temporal modeling*, v' is reshaped to $[N, (B \times L), D]$. We then project this into key and query matrices k and q and compute v'' using our *Temporal Attentive Soft Token Merging* as follows:

$$k = W_{\text{key}}(v'.\text{reshape}(N, B \times L, D)) \tag{5.2}$$

$$q = W_{\text{query}}(v'.\text{reshape}(N, B \times L, D)) \tag{5.3}$$

$$v'' = v' + \text{softmax}(\text{matmal}(q, k)) \cdot v' \tag{5.4}$$

The softmax operation models temporal weights and *softly* merges tokens along the temporal dimension. This is distinct from spatial token merging, which employs average pooling and reduces the token count. Here, we use a weighted average pooling along the temporal dimension, maintaining the original token count.

Our approach, depicted in Figure 5.1, maintains the integrity of pre-existing, well-trained image-text models, thus avoiding the need for model realignment, a requirement in alternative designs that insert a temporal Q-former between the visual encoder and the language model.

We proceed to evaluate the performance of fine-tuned SimVLG-Video models in video captioning tasks, utilizing OPT-2.7b as the language model decoder. Our investigation includes two specific variants of SimVLG-Video: the first is exclusively pre-trained on image data, while the second is further enhanced by pre-training on a corpus of 2 million video-text pairs sourced from the WebVid [8] dataset. To provide a comprehensive evaluation, we benchmark SimVLG-Video against five distinct models, described as follows:

- **Baseline (concat)**: This model processes each frame of a video individually and concatenates their visual features to generate a single prompt for the LLM. This method is analogous to the strategy employed in InstructBLIP.
- **Baseline (mean)**: Similar to the concat baseline, this model processes each video frame individually but averages the visual features to create a single prompt for the LLM.
- **Video-LLaMA**: This variant incorporates the BLIP-2 framework and enhances it with an additional temporal Q-former layer. For this evaluation, we focus solely on the vision-language component of Video-LLaMA.
- **VideoChat**: This model extends BLIP-2 by integrating additional Uniformer modules within the ViT architecture and also incorporates learnable temporal queries in its Q-former component.
- **VideoCoCa**: In this model, we adapt the OpenCoCa framework by mlfoundations and augment the existing CoCa architecture with a learnable attentional pooler, resulting in VideoCoCa.

Table 5.1: Comparison of different models' performance on MSR-VTT captioning. Models are pre-trained using 2 million video-text pairs from WebVid dataset, except for image pre-trained SimVLG.

Models	Image pre-trained	Video pre-trained	CIDEr	BLEU-4	METEOR	ROUGE
Baseline (concat)	✓	✓	65.5	44.4	31.9	64.1
Baseline (mean)	✓	✓	67.8	47.3	32.2	65.0
SimVLG-video	✓		68.4	47.6	32.4	65.3
SimVLG-video	✓	✓	69.8	48.3	32.6	65.8
SimVLG-video-scst	✓	✓	**74.0**	**49.2**	**33.0**	**66.5**
Video-LLaMA	✓	✓	59.3	47.7	29.6	63.7
VideoChat	✓	✓	58.0	46.5	29.5	63.4
VideoCoCa	✓	✓	63.0	48.5	31.4	64.8

Evaluation on MSR-VTT As detailed in Table 5.1, SimVLG-Video demonstrates superior performance relative to the baseline models, even without the aid of video-text pre-training. This result highlights the effectiveness of our proposed *Temporal Attentive Soft Token Merging* in capturing temporal dynamics. Additionally, we observe an enhancement in performance when incorporating video-text pre-training along with Self-Critical Sequence Training (SCST) [163].

Temporal Attentive Soft Token Merging has the distinct advantage of maintaining the integrity of the well-pretrained image-text model (i.e., SimVLG-Image). This contrasts with models such as Video-LLaMA and VideoChat, where the original BLIP-2 architecture is altered, necessitating a complex re-alignment process using video-text pairs. Our empirical analysis indicates that such re-alignment is a non-trivial endeavor. It is worth noting that our VideoCoCa model is at a disadvantage when benchmarked against Google's reported results, which benefit from extensive training on a much larger dataset of image-text and video-text pairs.

Evaluation on MSVD Similarly, we evaluate SimVLG's performance against Video-LLaMA, VideoChat, and VideoCoCa using the MSVD video captioning dataset, which

Table 5.2: Comparison of different models' performance on MSVD captioning.

Models	CIDEr	BLEU-4	METEOR	ROUGE
Video-LLaMA	121.2	61.6	40.3	77.8
VideoChat	118.4	64.1	41.0	78.7
VideoCoCa	150.9	67.7	45.3	81.9
SimVLG-video	**158.2**	**68.4**	**46.8**	**83.1**

is presented in Table 5.2. Our results corroborate that SimVLG consistently surpasses these competing models, further attesting to its robust performance across different video captioning tasks.

5.1.3. Summary

In summary, we have also extended SimVLG's applicability to video captioning tasks by incorporating the *Temporal Attentive Soft Token Merging* into its ViT. This enhances the model's temporal modeling capabilities, culminating in the creation of SimVLG-Video. This extension has proven efficacious, delivering commendable performance even without specialized video-text pre-training. Our investigation underscores that a temporal module, which does not disrupt the integrity of the well-pretrained image-text model (e.g., BLIP-2 and SimVLG), is a key factor contributing to this success.

Section 5.2

Effective Adaptation of VMs

In this section, we present Label Hallucination for Few-shot Classification (LabelHalluc) [77]. The work has been published in AAAI 2022. In this work, we explore a straightforward yet highly effective approach for adapting pre-trained vision models when provided with only a limited number of examples.

5.2.1. Overview and Motivation

Overview Few-shot classification requires adapting knowledge learned from a large annotated base dataset to recognize novel unseen classes, each represented by few labeled examples. In such a scenario, pretraining a network with high capacity on the large dataset and then finetuning it on the few examples causes severe overfitting. At the same time, training a simple linear classifier on top of "frozen" features learned from the large labeled dataset fails to adapt the model to the properties of the novel classes, effectively inducing underfitting. In this chapter we propose an alternative approach to both of these two popular strategies. First, our method pseudo-labels the entire large dataset using the linear classifier trained on the novel classes. This effectively "hallucinates" the novel classes in the large dataset, despite the novel categories not being present in the base database (novel and base classes are disjoint). Then, it finetunes the entire model with a distillation loss on the pseudo-labeled base examples, in addition to the standard cross-entropy loss on the novel dataset. This step effectively trains the network to recognize contextual and appearance cues that are useful for the novel-category recognition but using the entire large-scale base dataset and thus overcoming the inherent data-scarcity problem of few-shot learning. Despite the simplicity of the approach, we show that that our method outperforms the state-of-the-art on four well-established few-shot classification benchmarks.

Motivation Deep learning has emerged as the prominent learning paradigm for large data scenarios and it has achieved impressive results in wide range of application domains, including computer vision [102], NLP [37] and bioinformatics [172]. However, it remains difficult to adapt deep learning models to settings where few labeled examples are available, since large-capacity models are inherently prone to overfitting.

Few-shot learning is usually studied under the episodic learning paradigm, which simulates the few-shot setting during training by repeatedly sampling few examples

from a small subset of categories of a large base dataset. Meta-learning algorithms [49, 100, 158, 183, 201] optimized on these training episodes have advanced the field of few-shot classification. However, recent works [24, 38, 196] have shown that a pure transfer learning strategy is often more competitive. For example, Tian et al. [196] proposed to first pretrain a large capacity classification model on the base dataset and then to simply learn a linear classifier on this pretrained representation using the few novel examples. The few-shot performance of the transferred model can be further improved by multiple distillation iterations [52], or by combining several losses simultaneously, e.g., entropy maximization, rotational self-supervision, and knowledge distillation [155].

In this chapter, we follow the transfer learning approach. However, instead of freezing the representation to the features learned from the base classes [155, 165, 196], we finetune the entire model. Since finetuning the network using only the few examples would result in severe overfitting (as evidenced by our ablations), we propose to optimize the model by re-using the entire base dataset but only after having swapped the original labels with pseudo-labels corresponding to the novel classes. This is achieved by running on the base dataset a simple linear classifier trained on the few examples of the novel categories. The classifier effectively "hallucinates" the presence of the novel classes in the base images. Although we empirically evaluate our approach in scenarios where the classes of the base dataset are completely disjoint from the novel categories, we demonstrate that this large-scale pseudo-labeled data enables effective finetuning of the entire model for recognition of the novel classes. The optimization is carried out using a combination of distillation over the pseudo-labeled base dataset and cross-entropy minimization over the few-shot examples. The intuition is that although the novel classes are not properly represented in the base images, many base examples may include objects that resemble those of the novel classes as encoded by

the soft pseudo-labels that define the probabilities of belonging to the novel classes. For example, the pseudo-labeling may assign a probability of 0.6 for a base image of a tiger to belong to the novel class "domestic cat" given their appearance similarities. Or it may assign large novel-class pseudo-label probability to a base images because its true base category shares similar contextual background with the novel class, such in the case of "cars" and "pedestrians" which are both likely to appear in street scenes. Fine-tuning the entire model on these soft pseudo-labels using a distillation objective (combined with the cross-entropy loss on the few novel image examples) trains the network to recognize these similar or contextual cues on the base dataset, thus steering the representation towards features that are useful for the recognition of the novel classes. Furthermore, because the base dataset is large-scale, these examples serve the role of massive non-parametric data augmentation yielding a representation that is quite general and does not overfit, thus overcoming the data scarcity problem inherent in few-shot learning. An overview of our proposed approach is provided in Fig. 5.2.

We note that pseudo-labeling has been widely used before for semi-supervised learning where the unlabeled examples belong to the same classes as the labeled ones [23, 150, 184]. Pseudo-labeling has also been adapted to the few-shot setting [106, 209] but still under the empirical setting where novel classes are contained in the unlabeled dataset. The novelty of our work lies in showing that the advantages of pseudo-labeling extend even to the extreme setting where the set of base classes and the set of novel classes are completely disjoint. We also note that our work differs from transductive few-shot learning [38, 209] which requires the testing set of unlabeled examples used during the training. Instead, our method operates in a pure inductive setting where within each episode only the small set of novel labeled examples and the base dataset are used for finetuning.

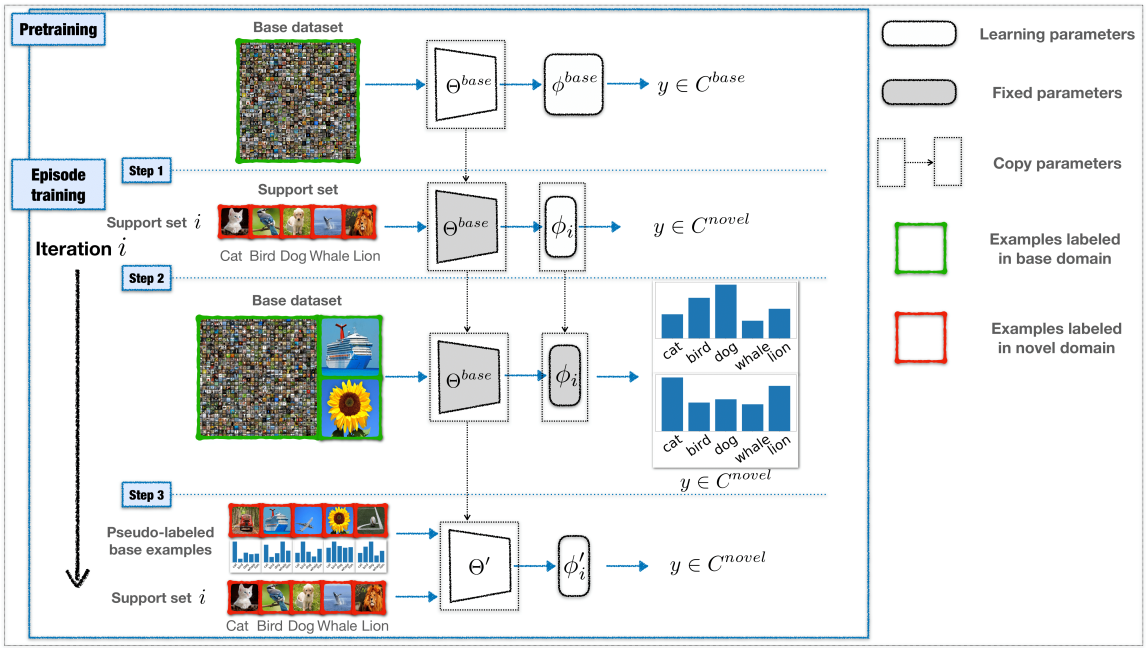

Figure 5.2: Overview of our proposed approach in an illustrative setting involving 1-shot classification of 5 novel classes. *Pretraining* learns the backbone model Θ and a classification head ϕ_0 from a labeled base dataset. The backbone is used to compute embeddings for the subsequent stages, while the classification head is discarded. During *Episode training*, step 1) learns a linear classifier ϕ_1 in the novel domain using the support set and the fixed embedding Θ. Step 2) pseudo-labels the base dataset with respect to the label space of the novel domain using the fixed embedding Θ and the classifier ϕ_1. Step 3) re-learns both the embedding and the classifier with the support set and the pseudo-labeled base dataset using a combination of distillation and cross-entropy maximization. Note that the base dataset and the support set do not share any classes.

5.2.2. Methods and Experiments

Problem statement We now formally define the few-shot classification problem considered in this work. We adopt the common setup which assumes the existence of a large scale labeled base dataset used to discriminatively learn a representation useful for the subsequent novel-class recognition. Let $\mathcal{D}^{base} = \{x_t^{base}, y_t^{base}\}_{t=1}^{N^{base}}$ be the base dataset, with label $y_t^{base} \in \mathcal{C}^{base}$. It is assumed that both the number of classes ($|\mathcal{C}^{base}|$) and the number of examples (N^{base}) are large in order to enable good representation learning. We denote with $\mathcal{D}^{novel} = \{x_t^{novel}, y_t^{novel}\}_{t=1}^{N^{novel}}$ the novel

dataset, with $y_t^{novel} \in \mathcal{C}^{novel}$. The base classes and novel classes are disjoint, i.e., $\mathcal{C}^{base} \cap \mathcal{C}^{novel} = \varnothing$. We assume the training and testing of the few-shot classification model to be organized in episodes. At each episode i, the few-shot learner is given a support set $\mathcal{D}_i^{support} = \{x_{i,t}^{support}, y_{i,t}^{support}\}_{t=1}^{NK}$ involving K novel classes and N examples per class sampled from $\mathcal{D}^{novel}$ (with N being very small, typically ranging from 1 to 10). The learner is then evaluated on the query set $\mathcal{D}_i^{query} = \{x_{i,t}^{query}, y_{i,t}^{query}\}_{t=1}^{N'K}$, which contains examples of the same K classes as those in $\mathcal{D}_i^{support}$. Thus, the query/support sets serve as few-shot training/testing sets, respectively. At each episode i, the few-shot learner adapts the representation/model learned from the large-scale $\mathcal{D}^{base}$ to recognize the novel classes given the few training examples in $\mathcal{D}_i^{support}$.

Learning the embedding representation on the base dataset We first aim at learning from the base dataset an embedding model that will transfer and generalize well to the downstream few-shot problems. We follow the approach of Tian et al. [196] (denoted as RFS) and train discriminatively a convolutional neural network consisting of a backbone f_Θ and a final classification layer g_ϕ. The parameters $\{\Theta, \phi\}$ are optimized jointly for the $\left|\mathcal{C}^{base}\right|$-way base classification problem using the dataset $\mathcal{D}^{base}$:

$$\Theta^{base}, \phi^{base} = \text{argmin}_{\Theta,\phi} \mathbb{E}_{\{x,y\}\in\mathcal{D}^{base}} \mathcal{L}_{CE}(g_\phi(f_\Theta(x)), y) \tag{5.5}$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss.

Prior work has shown the quality of the embedding representation encoded by parameters Θ^{base} can be further improved by knowledge distillation [196], rotational self-supervision [155] or by enforcing representations equivalent and invariant to sets of image transformations [165]. In the experiements presented in this section, we follow the embedding learning strategies of SKD [155] and IER [165]. However, note that

our approach is independent of the specific method used for embedding learning.

Hallucinating the presence of novel classes in the base dataset In order to pseudo-label the base dataset according to the novel classes, we first train a classifier on the support set. For each episode i in the meta-learning phase, we learn a linear classifier ϕ_i on top of the *fixed* feature embedding model Θ^{base} using the few-shot support set $\mathcal{D}_i^{support} = \{x_{i,t}^{support}, y_{i,t}^{support}\}_{t=1}^{NK}$.

$$\phi_i = \text{argmin}_\phi \mathbb{E}_{\{x,y\}\in\mathcal{D}_i^{support}} \mathcal{L}_{CE}(g_\phi(f_{\Theta^{base}}(x)), y) \tag{5.6}$$

Note that in previous works [155, 165, 196], ϕ_i is directly evaluated on query set $\mathcal{D}_i^{query}$ to produce the final few-shot classification results. Instead here we use the resulting model $g_{\phi_i}(f_{\Theta^{base}}(x))$ to re-label the base dataset according to the ontology of the novel classes in episode i. We denote with $\hat{y}_{i,t}^{base}$ the vector of logits (the outputs before the softmax) generated by applying the learned classifier to example x_t^{base}, i.e., $\hat{y}_{i,t}^{base} = g_{\phi_i}(f_{\Theta^{base}}(x_t))$ for $t = 1, \ldots, N^{base}$. These soft pseudo-labels are used to retrain the full model via knowledge distillation, as discussed next.

Finetuning the whole model to recognize novel classes We finally finetune the whole model (i.e., the backbone and the classifier) using mini-batches containing an equal proportion of support and base examples. The loss function for the base examples is knowledge distillation [70], while the objective minimized for the support examples is the cross-entropy (CE). In other words, we optimize the parameters of the model on a mixing of the two losses:

$$\begin{aligned}\Theta_i', \phi_i' = \text{argmin}_{\Theta,\phi} \lambda_{KL} \mathbb{E}_{\{x,y\}\in\mathcal{D}^{base}} \mathcal{L}_{KL}(g_\phi(f_\Theta(x)), \hat{y}) + \\ \lambda_{CE} \mathbb{E}_{\{x,y\}\in\mathcal{D}_i^{support}} \mathcal{L}_{CE}(g_\phi(f_\Theta(x)), y) \ .\end{aligned} \tag{5.7}$$

where $\hat{y}$ denotes the hallucinated pseudo-label, $\mathcal{L}_{KL}$ is the KL divergence between the predictions of the model and the pseudo-labels scaled by temperature T, and $\{\lambda_{KL}, \lambda_{CE}\}$ are hyper-parameters trading off the importance of the two losses. Since the support set is quite small (under certain experimental settings in each episode we have five novel classes and only one example for each novel class), we use data augmentation to generate multiple views of each support image, so as to obtain enough examples to fill half of the mini-batch.

Finally, the resulting model $g_{\phi'_i}(f_{\Theta'_i}(x))$ is evaluated on the query set $\mathcal{D}_i^{query} = \{x_t^{novel}, y_t^{novel}\}_{t=1}^{N'K}$. The final results are reported by averaging the accuracies of all episodes.

We note that although the operations of pseudo-labeling and finetuning are presented as separate and in sequence, in practice for certain datasets we found more efficient to generate the target pseudo-labels on the fly for the base examples loaded in the mini-batch without having to store them on disk.

Datasets We evaluate our method on four widely used few-shot recognition benchmarks: miniImageNet [201], tieredImageNet [161], CIFAR-FS [12], and FC100 [143].

miniImageNet [201] is one of the most commonly used benchmark for few-shot classification. It is a subset of ImageNet [36] containing images downsampled to a resolution of 84×84 pixels. It includes 100 classes, each class with 600 examples. It is further divided into 64 classes for meta-training, 16 classes for meta-validation and 20 classes for meta-testing. **tieredImageNet** [161] is another subset of ImageNet. It contains a total of 608 classes, with 351 classes used for meta-training, 97 classes for meta-validation and the remaining 160 classes for meta-testing. **CIFAR-FS** [12] is derived from CIFAR-100 dataset. The original 100 classes are split into 64 classes for meta-training, 16 classes for meta-validation and 20 classes for meta-testing. **FC-100** [143] is also obtained from CIFAR-100. It has 60 classes for meta-training, 20

classes for meta-validation and 20 classes for meta-testing.

Experimental setup Network Architecture. To make fair comparison to recent works, we adopt the popular ResNet-12 [68] as our backbone. The network has 4 residual blocks, each containing 3 convolutional layers with 3×3 convolution. A 2×2 max-pooling is applied at the end of each of the first three blocks and an average-pooling is used in the last one. Our ResNet-12 is identical to those used in RFS [196], SKD [155] and IER [165]. Thus the number of channels for 4 residual block is set to (64,160,320,640).

Optimization details. For the embedding training, we use the public code implementations of SKD [155] and IER [165]. For the training of the linear classifiers on top of frozen features, we use LogisticRegression with the LBFGS optimizer from scikit-learn package [16], as in RFS [196]. When finetuning, we use the SGD optimizer with a momentum of 0.9 and a weight decay of $5e^{-4}$ across all experiments in the four benchmarks. The learning rate for the network up to the penultimate layer is set to 0.025 while the final classification layer uses a learning rate of 0.05.

We use mini-batches of 250 examples. For the 5-way 5-shot classification in miniImageNet, CIFAR-FS and FC100, we generate 5 views from each of the 25 novel images in the support in order to obtain 125 examples. We use the same data augmentation transformations employed in prior works [155, 165, 196]). We fill the remaining half the mini-batch by sampling 125 distinct examples from the base dataset with associated pseudo labels. The finetuning runs for 1 epoch in order to complete a pass over the entire base dataset for each episode. This amount to $\sim$300 steps. Similarly in 5-way 1-shot classification, each mini-batch has a support set of 125 novel examples (5 distinct images augmented 25 times) and 125 pseudo-labeled base examples.

The tieredImageNet dataset is much larger than the other three datasets and

		miniImageNet 5-way		tieredImageNet 5-way	
model	**backbone**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
ProtoNet [183]	ResNet-12	60.37 ± 0.83	78.02 ± 0.57	65.65 ± 0.92	83.40 ± 0.65
TADAM [143]	ResNet-12	58.50 ± 0.30	76.70 ± 0.30	-	-
TapNet [227]	ResNet-12	61.65 ± 0.15	76.36 ± 0.10	63.08 ± 0.15	80.26 ± 0.12
MetaOptNet [109]	ResNet-12	62.64 ± 0.61	78.63 ± 0.46	65.99 ± 0.72	81.56 ± 0.53
MTL [191]	ResNet-12	61.20 ± 1.80	75.50 ± 0.80	65.62 ± 1.80	80.61 ± 0.90
Shot-Free [159]	ResNet-12	59.04 ± 0.43	77.64 ± 0.39	66.87 ± 0.43	82.64 ± 0.43
DSN-MR [180]	ResNet-12	64.60 ± 0.72	79.51 ± 0.50	67.39 ± 0.83	82.85 ± 0.56
DeepEMD [235]	ResNet-12	65.91 ± 0.82	82.41 ± 0.56	71.16 ± 0.87	86.03 ± 0.58
FEAT [226]	ResNet-12	66.78 ± 0.20	82.05 ± 0.14	70.80 ± 0.23	84.79 ± 0.16
Neg-Cosine [124]	ResNet-12	63.85 ± 0.81	81.57 ± 0.56	-	-
RFS-simple [196]	ResNet-12	62.02 ± 0.63	79.64 ± 0.44	69.74 ± 0.72	84.41 ± 0.55
RFS-distill [196]	ResNet-12	64.82 ± 0.82	82.41 ± 0.43	71.52 ± 0.69	86.03 ± 0.49
AssoAlign [1]	ResNet-18†	59.88 ± 0.67	80.35 ± 0.73	69.29 ± 0.56	85.97 ± 0.49
AssoAlign [1]	WRN-28-10‡	65.92 ± 0.60	82.85 ± 0.55	74.40 ± 0.68	86.61 ± 0.59
SKD-GEN1 [155]	ResNet-12	66.54 ± 0.97§	83.18 ± 0.54§	72.35 ± 1.23§	85.97 ± 0.63§
P-Transfer [176]	ResNet-12	64.21 ± 0.77	80.38 ± 0.59	-	-
InfoPatch [57]	ResNet-12	67.67 ± 0.45	82.44 ± 0.31	71.51 ± 0.52	85.44 ± 0.35
MELR [48]	ResNet-12	67.40 ± 0.43	83.40 ± 0.28	72.14 ± 0.51	87.01 ± 0.35
IEPT [239]	ResNet-12	67.05 ± 0.44	82.90 ± 0.30	72.24 ± 0.50	86.73 ± 0.34
IER-distill [165]	ResNet-12	66.85 ± 0.76§	84.50 ± 0.53§	72.74 ± 1.25§	86.57 ± 0.81§
Ours w/ SKD	ResNet-12	67.50 ± 1.01	85.60 ± 0.52	72.80 ± 1.20	86.93 ± 0.60
Ours w/ IER	ResNet-12	**68.28 ± 0.77**	**86.54 ± 0.46**	**73.34 ± 1.25**	**87.68 ± 0.83**

Table 5.3: Comparison of our method (Label-Halluc) against the state-of-the-art on miniImageNet and tieredImageNet. We report our results with 95% confidence intervals on meta-testing split of miniImageNet and tieredImageNet. Training is done on the training split only. † indicates using a higher resolution of training images. ‡ indicates a larger model than ResNet-12. § indicates our implementations. This makes the fairest comparisons to ours by allowing that those methods are evaluated on exact same episodes.

the images have much higher resolution. Thus, a finetuning procedure that iterates through the whole base dataset for each episode is intractable. Thus, we finetune for 200 steps for both the 1-shot and the 5-shot settings.

Each of these four benchmarks includes a meta-training set (the base dataset), a meta-validation set and a meta-testing set (the novel-class dataset) organized in episodes. The meta-validation set is only used for searching hyper-parameters.

The experiments are carried out on a desktop server with Intel i9-9960X CPU and four NVIDIA RTX-2080Ti GPUs.

Results on ImageNet-based few-shot benchmarks Table 5.3 provides a comparison between our approach and the state-of-the-art in few-show classification on the

		CIFAR-FS 5-way		FC-100 5-way	
model	**backbone**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
ProtoNet [183] (NIPS'17)	ResNet-12	72.2 ± 0.7	83.5 ± 0.5	37.5 ± 0.6	52.5 ± 0.6
TADAM [143] (NIPS'18)	ResNet-12	-	-	40.1 ± 0.4	56.1 ± 0.4
MetaOptNet [109] (CVPR'19)	ResNet-12	72.6 ± 0.7	84.3 ± 0.5	41.1 ± 0.6	55.5 ± 0.6
MTL [191] (CVPR'19)	ResNet-12	-	-	45.1 ± 1.8	57.6 ± 0.9
Shot-Free [159] (ICCV'19)	ResNet-12	69.2 ± n/a	84.7 ± n/a	-	-
DSN-MR [180] (CVPR'20)	ResNet-12	75.6 ± 0.9	86.2 ± 0.6	-	-
DeepEMD [235] (CVPR'20)	ResNet-12	-	-	46.5 ± 0.8	63.2 ± 0.7
RFS-simple [196] (ECCV'20)	ResNet-12	71.5 ± 0.8	86.0 ± 0.5	42.6 ± 0.7	59.1 ± 0.6
RFS-distill [196] (ECCV'20)	ResNet-12	73.9 ± 0.8	86.9 ± 0.5	44.6 ± 0.7	60.9 ± 0.6
AssoAlign [1] (ECCV'20)	ResNet-18‡	-	-	45.8 ± 0.5	59.7 ± 0.6
SKD-GEN1 [155] (Arxiv'20)	ResNet-12	76.6 ± 0.9§	88.6 ± 0.5§	46.5 ± 0.8§	64.2 ± 0.8§
InfoPatch [57] (AAAI'21)	ResNet-12	-	-	43.8 ± 0.4	58.0 ± 0.4
IER-distill [165] (CVPR'21)	ResNet-12	77.6 ± 1.0§	89.7 ± 0.6§	48.1 ± 0.8§	65.0 ± 0.7§
Label-Halluc (pretrained w/ SKD)	ResNet-12	77.3 ± 0.9	89.5 ± 0.5	47.3 ± 0.8	67.2 ± 0.8
Label-Halluc (pretrained w/ IER)	ResNet-12	**78.0 ± 1.0**	**90.5 ± 0.6**	**49.1 ± 0.8**	**68.0 ± 0.7**

Table 5.4: Comparison of Label-Halluc (ours) to prior works on CIFAR-FS and FC-100. We report our results with 95% confidence intervals on meta-testing split of CIFAR-FS and FC-100. Training is done on the training split only. ‡ indicates a different model. § indicates our implementations.

two ImageNet-based few-shot benchmarks. Our method is denoted as Label-Halluc. On **miniImageNet**, our method using the SKD pretraining of the backbone yields an absolute improvement of 0.96% over SKD-GEN1 in the one-shot setting. The improvement become more substantial under the 5-shot setting, with our method producing a gain of 2.42% over SKD-GEN1. When pretrained with IER [165], our approach achieves one-shot classification accuracy of 68.28 ± 0.77, which is over 1.4% better than all reported results. Under the 5-shot setting, our method improves by 2.04% over IER-distill which had the best reported number, yielding a new state-of-the-art accuracy of 86.54%. On the **tieredImageNet** benchmark, our method pretrained with SKD performs on par with concurrent works [48, 239] and outperforms SKD [155] by 0.45% under the 1-shot setting and by 0.96% under the 5-shot setting. When pretrained with IER, our approach improves over IER-distill by 0.60% and 1.11% under the 1-shot and 5-shot settings, respectively, yielding a new state-of-the-art even for this benchmark.

Results on CIFAR-based few-shot benchmarks Table 5.4 compares our method, Label-Halluc, against the state-of-the-art on the two CIFAR-based few-shot benchmarks. On **CIFAR-FS**, the improvements over SKD-GEN1 (our implementation) for 1-shot and 5-shot are 0.7% and 0.9%, respectively. Note that these gains derive exclusively from the addition of the distillation over pseudo-labeled base examples. When using IER-distill as embedding learning, our method improves the baseline by 0.4% and 0.8% in the 1-shot and the 5-shot settings, respectively. On **FC100**, our method achieves improves over the best reported numbers by 0.8% and 3.0% in the 1-shot and 5-shot setting, respectively, when pretrained with SKD. The improvements are 1.0% and 3.0% when pretrained with IER.

5.2.3. Summary

We propose the simple strategy of label hallucination to enable effective finetuning of large-capacity models from few-shot examples of the novel classes. We demonstrate that even in the extreme scenario where the labels of the base dataset and the labels of the novel examples are completely disjoint, this simple procedure improves over the popular strategies of transfer learning via finetuning on the novel examples or via linear classification on top of a frozen representation. Results on four well-established few-shot classification benchmarks show that our method outperforms the state-of-the-art.

Section 5.3

Effective Adaptation of LMs

In this section, we present "Contrastive Learning for Prompt-based Few-shot Language Learners" (LM-SupCon) [80]. The work has been published in NAACL 2022. This work proposed a simple yet effective method for fine-tuning LLMs given a few in-context examples.

5.3.1. Overview and Motivation

Overview The impressive performance of GPT-3 using natural language prompts and in-context learning has inspired work on better fine-tuning of moderately-sized models under this paradigm. Following this line of work, we present a contrastive learning framework that clusters inputs from the same class for better generality of models trained with only limited examples. Specifically, we propose a supervised contrastive framework that clusters inputs from the same class under different augmented "views" and repel the ones from different classes. We create different "views" of an example by appending it with different language prompts and contextual demonstrations. Combining a contrastive loss with the standard masked language modeling (MLM) loss in prompt-based few-shot learners, the experimental results show that our method can improve over the state-of-the-art methods in a diverse set of 15 language tasks. Our framework makes minimal assumptions on the task or the base model, and can be applied to many recent methods with little modification.

Motivation The prompt-based fine-tuning method reduces the gap between pre-training and fine-tuning by forming the fine-tuning task into a masking language problem. A language prompt is a piece of text appended to the query input enabling the model to come up with better predictions [169, 194]. For instance, by feeding a language model with *"The story is not worth reading, a truly ___ one."*, the model assigns a higher probability for the blank to be filled with *"terrible"* than *"great"*. Here, *"a truly ___ one."* is called the template of the prompt and *"terrible"* or *"great"* is the label word. Recently, LM-BFF [55] shows that appending demonstrations (*e.g."This is an amazing movie, a truly great one"*) to inputs can help the model to better understand the label word, leading to further improved results.

In this work, we show that Supervised Contrastive Learning (SupCon) [97] at the feature space can be beneficial during the *fine-tuning* of prompt-based few-shot

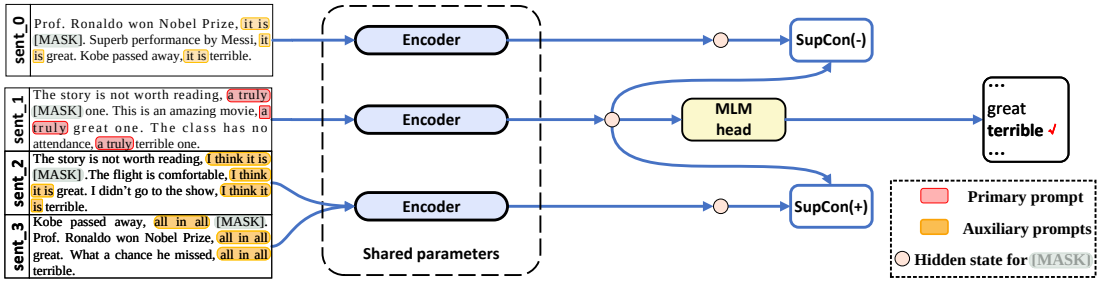

Figure 5.3: Overview of our proposed method. Besides the standard prompt-base MLM loss on label words "great" and "terrible", we introduce a SupCon loss on multi-views of input text. The positive pair is sentences (with sampled templates and/or demonstrations) in the same class, e.g. sent_1 and sent_3, or itself with a different template and demonstrations, e.g. sent_1 and sent_2. The negative sentence pair is input sentences (with sampled templates and/or demonstrations) in different classes, e.g. sent_1 and sent_0.

language learners, with proper data augmentation.

Data augmentation is the key component of SupCon. While there exists many augmentation techniques like Cutmix [230], Mixup [238] in computer vision and EDA [214], AEDA [95] for text, data augmentation remains challenging.

However, prompt-based few-shot learners with demonstrations actually provide us with a natural way to create multiple "views" (augmentations) of a single example, *i.e.*, for a fixed set of label words, we can sample different templates and different demonstrations to append to the input text (shown in Figure 5.3). This allows us to construct diverse input texts that are consistent and complete. By applying SupCon to cluster the above two example inputs with very different contents but the same label, our method is able to obtain an additional supervision at the feature space which is crucial if we are only given a few labeled examples.

The main contributions of ours are: (1) A Supervised Contrastive Learning framework for prompt-based few-shot learners. (2) An effective data augmentation method using prompts for contrastive learning with prompt-based learners.

5.3.2. Methods and Experiments

Problem formulation. Following the few-shot setting in LM-BFF, we assume to have access to a pre-trained language model $\mathcal{M}$, datasets $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ with label space $\mathcal{Y}$. There are only $K = 16$ examples per class in $\mathcal{D}_{\text{train}}$.

Fine-tuning with prompts and demonstrations. Prompt-based methods treat a classification problem as a masked language modeling (MLM) problem. They take as input a sentence (sent) and a masked template (temp) (i.e., $x_{\text{prompt}} = \text{sent}, \text{temp}([\text{mask}])$), and find the best token to fill in the [mask]. This leads to a MLM loss $\mathcal{L}_{\text{MLM}} = \text{MLM}(x_{\text{prompt}}, y)$, where y is the label word corresponding to x_{prompt}. LM-BFF [55] further appends demonstrations of label words to improve the results: $x_{\text{prompt+demo}} = \text{sent}_0, \text{temp}_0([\text{mask}]), \text{sent}_i, \text{temp}_0(\text{word}_i)$, where word_i is the label word for sent_i, and sent_i is sampled from the training set. Then the classification loss becomes:

$$\mathcal{L}_{\text{MLM}} = \text{MLM}(x_{\text{prompt+demo}}, y) \tag{5.8}$$

Language-based Supervised Contrastive Loss. For applying SupCon on multi-views of an input text, we need to first obtain two views of a text:

$$\begin{aligned} x_1 =& \text{sent}_0, \text{temp}_0([\text{mask}]), \text{sent}_i, \text{temp}_0(\text{word}_i) \\ x_2 =& \text{sent}_0, \text{temp}_j([\text{mask}]), \text{sent}_k, \text{temp}_j(\text{word}_k) \end{aligned}$$

where x_1 is identical to $x_{\text{prompt+demo}}$ in LM-BFF. We sample a new template (temp_j), demonstration (sent_k) and the corresponding label word (word_k) to replace those in x_1, to create a second view of input x_2. With x_1 and x_2, we can compute SupCon loss.

Task	LM-BFF	LM-BFF + ours	PET	PET + ours
SST-2 (acc)	89.2 (1.3)	**90.6** (0.1)	88.4 (1.0)	**89.9** (0.6)
Subj (acc)	88.6 (3.3)	**90.4** (1.1)	89.2 (1.5)	**90.6** (1.6)
SST-5 (acc)	47.9 (0.8)	**49.5** (1.1)	46.0 (0.9)	**48.8** (1.2)
CoLA (Matt.)	6.1 (5.3)	**10.2** (5.8)	3.5 (3.4)	**5.9** (3.3)
TREC (acc)	82.8 (3.1)	**83.3** (1.5)	77.8 (9.1)	**82.3** (4.6)
MNLI (acc)	61.0 (2.1)	**64.0** (2.0)	58.2 (1.1)	**58.9** (3.1)
MNLI-mm (acc)	62.5 (2.1)	**65.5** (2.7)	59.8 (1.2)	**61.0** (3.3)
SNLI (acc)	66.9 (2.4)	**69.9** (2.4)	63.1 (2.5)	**65.7** (3.9)
QNLI (acc)	60.7 (1.7)	**66.4** (3.5)	61.5 (3.3)	**63.5** (3.7)
QQP (acc)	62.5 (2.6)	**68.8** (3.8)	61.9 (3.5)	**65.7** (4.3)
RTE (acc)	64.3 (2.7)	**65.1** (3.5)	60.9 (4.7)	**65.1** (3.5)
MRPC (F1)	75.5 (5.2)	**78.2** (3.1)	70.6 (6.0)	**75.7** (6.1)
MR (acc)	83.3 (1.4)	**85.8** (0.6)	85.0 (0.6)	**85.2** (0.9)
MPQA (acc)	83.6 (1.8)	**84.6** (1.5)	81.3 (2.6)	**81.8** (2.4)
CR (acc)	88.9 (1.0)	**89.4** (1.0)	89.3 (1.0)	**90.5** (0.5)

Table 5.5: Few-shot experiments of baseline methods and ours. LM-BFF is a prompt-based method with demonstrations of label words and PET is one without demonstrations. The experimental results show the means and standard deviations from 5 different train-test splits.

The total loss is then

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{SupCon}} \tag{5.9}$$

Computational overhead. Because $\mathcal{L}_{\text{SupCon}}$ requires one additional forward and backward pass in each tuning iteration, we observe that the training cost is raised by a factor 1.5 compared to the baselines.

Evaluation datasets and protocol. We evaluate our method on 15 classification tasks studied in LM-BFF and follow the same setup as them to allow fair comparisons. Contrastive learning algorithms benefit from large batch training. Thus, we report baselines with the same large batch size as ours.

Our method uses a single prompt/template (primary prompt) for the prediction

of each task, and a set of prompts (auxiliary prompts) for generating multi-views of inputs for contrastive learning. The auxiliary prompts can be either manually designed or generated by a searching algorithm. In this work, we use the top-20 generated prompts from LM-BFF's project page and we randomly sample templates in these 20 prompts to produce second views of our inputs. Unless otherwise noted, we apply *both* random templates and random demonstrations to create second views of inputs for the contrastive learning.

Main results on 15 tasks We use RoBERTa-base. We compare ours with LM-BFF (a method w/ demonstrations) and PET [169] (a method w/o demonstration).

Table 5.5 shows that our SupCon loss can consistently boost the performance of baseline prompt-based fine-tuning method LM-BFF. The introduction of SupCon loss has a maximum improvement of 6.3% in QQP and an average improvement of 2.5% across 15 tasks, likely due to the more generalized representations learned by SupCon. On average, the greater improvements by our model can be seen on the more difficult tasks

We want to emphasize that the input for baseline LM-BFF already appends different randomly sampled demonstrations at each tuning iteration. Thus, the improvement of our method can not be attributed to the diversity of inputs when learning from $\mathcal{L}_{\text{MLM}}$ of Equation 5.7, but to the $\mathcal{L}_{\text{SupCon}}$.

Table 5.5 also shows that our method works well even for prompt-based methods without demonstrations. PET, which is a method without demonstrations, works consistently worse than LM-BFF. However, with the additional SupCon loss, the few-shot performances of PET can be increased by an average of 2.3%. And the gap between having and not having demonstrations can be largely closed (see LM-BFF vs. PET+ours in Table 5.5). In some tasks, *e.g.*, SST-2, SST-5, QNLI, QQP, RTE MRPC, MR, and CR, the contribution of our SupCon loss can be even larger than

the sole use of the demonstrations for label words.

Task	LM-BFF	LM-BFF +Dec	LM-BFF +Dec +Lab	LM-BFF +ConCal	LM-BFF +ours
SST-2	89.2 (1.3)	90.1 (0.6)	**90.6** (0.5)	88.5 (2.0)	**90.6** (0.1)
Subj	88.6 (3.3)	87.3 (3.6)	88.4 (4.9)	83.8 (7.3)	**90.4** (1.1)
SST-5	47.9 (0.8)	47.2 (1.0)	46.5 (0.7)	47.9 (1.1)	**49.5** (1.1)
CoLA	6.1 (5.3)	9.8 (6.5)	7.2 (5.2)	6.7 (4.6)	**10.2** (5.8)
TREC	82.8 (3.1)	81.9 (3.0)	82.3 (3.0)	71.1 (7.0)	**83.3** (1.5)
MNLI	61.0 (2.1)	61.3 (2.1)	59.4 (1.3)	61.0 (0.8)	**64.0** (2.0)
-mm	62.5 (2.1)	63.2 (2.1)	61.4 (1.6)	62.5 (0.8)	**65.5** (2.7)
SNLI	66.9 (2.4)	67.0 (3.1)	65.8 (2.1)	67.0 (2.9)	**69.9** (2.4)
QNLI	60.7 (1.7)	60.0 (2.5)	60.2 (2.0)	60.9 (2.0)	**66.4** (3.5)
QQP	62.5 (2.6)	**69.0** (1.7)	65.4 (1.2)	62.2 (2.7)	68.8 (3.8)
RTE	64.3 (2.7)	**65.6** (1.5)	65.3 (2.4)	60.2 (1.9)	65.1 (3.5)
MRPC	75.5 (5.2)	69.4 (7.0)	66.5 (7.0)	**78.3** (3.1)	78.2† (3.1)
MR	83.3 (1.4)	85.0 (1.0)	84.6 (1.2)	84.0 (1.4)	**85.8** (0.6)
MPQA	83.6 (1.8)	82.3 (1.9)	84.3 (1.4)	72.3 (13.4)	**84.6** (1.5)
CR	88.9 (1.0)	89.3 (0.6)	**89.6** (0.7)	87.7 (1.1)	89.4 (1.0)

Table 5.6: Comparing our SupCon loss with Decoupling Label Loss (Dec), Label Condition Loss (Lab), and Contextual Calibration (ConCal). † We can achieve stronger performance 80.0 ± 1.8 by fixing templates/demonstrations when creating the second view of the input.

SupCon vs. other losses We further show that our method outperforms two latest methods that are designed to improve prompt-based language models. In ADAPET [194], the authors replace the traditional CrossEntropy loss with Decoupling Label Loss and Label Condition Loss in the prompt-based fine-tuning method PET, without demonstrations. Contextual Calibration [245] calibrates the output probabilities by considering context-free inputs, *i.e.*, " " or "N/A".

From Table 5.6 we observe that on 12 tasks our $\mathcal{L}_{\text{SupCon}}$ outperforms the other losses, while performs on-par in other tasks. Contextual Calibration does not achieve good results overall. We speculate two reasons for this. First, Contextual Calibration is designed for large models without fine-tuning like GPT (zero-shot setting). Second, the form of in-context learning in Contextual Calibration is different from the

Task	LM-BFF +ours	LM-BFF ensemble
SST-5 (acc)	**49.5** (1.1)	48.0 (0.8)
CoLA (Matt.)	**10.2** (5.8)	7.5 (4,7)
MNLI (acc)	**63.3** (2.4)	62.2 (1.8)
MNLI-mm (acc)	**65.1** (2.4)	64.0 (1.8)
QNLI (acc)	**66.4** (3.5)	63.8 (2.7)
MR (acc)	**85.8** (0.6)	85.7 (0.7)

Table 5.7: Comparing our single model trained with SupCon loss to an ensemble of 20 models.

demonstrations we study here.

Ensemble vs. our single model Our method uses 20 generated templates (auxiliary prompts) to construct multi-views of input sentences. But only a single prompt (primary prompt) and one set of label words are used for main predictions. Thus, there is only a single model from our method. Here, we compare our model to an ensemble comprised of 20 models trained separately with the 20 prompts. From Table 5.7, we find that our method even outperforms the ensemble with 20× more number of parameters, showing that it is a more efficient way to make use of the generated prompts. We speculate that because of the over-fitting nature of few-shot learners, members in the ensemble fail to produce substantial diverse prediction distributions.

Ablations: Input augmentation The success of contrastive learning heavily relies on the data augmentation. Our method takes advantage of prompt-based language learners and naturally creates multi-views of a single input by appending it with different templates and/or demonstrations. Compared to EDA which includes synonym replacement (SR), random insertion (RI), random swap (RS) and random deletion (RD), our strategy for augmentation does not lead to incomplete and inconsistent sentences, while introducing adequate variations for effective learning.

Task	LM-BFF	SR	RI	RS	RD	EDA	ours
SST-2	89.2	90.7	**90.8**	90.7	90.7	90.5	90.6
Subj	88.6	90.6	90.8	**91.0**	90.5	89.1	90.4
SST-5	47.9	47.7	49.2	48.2	47.9	46.7	**49.5**
CoLA	6.1	5.8	6.5	4.9	4.0	3.9	**10.2**
TREC	82.8	78.1	80.7	79.0	80.7	80.6	**83.3**
MNLI	61.0	61.8	62.4	61.0	58.1	58.9	**64.0**
-mm	62.5	63.6	64.8	62.7	60.3	60.9	**65.5**
SNLI	66.9	63.1	66.4	67.2	65.2	62.2	**69.9**
QNLI	60.7	65.3	65.3	**67.4**	64.8	62.5	66.4†
QQP	62.5	64.5	65.8	68.0	63.2	61.0	**68.8**
RTE	64.3	61.4	61.4	61.3	62.1	61.1	**65.1**
MRPC	75.5	77.6	77.7	**79.3**	78.7	79.1	78.2†
MR	83.3	85.5	85.5	85.5	85.3	85.6	**85.8**
MPQA	83.6	82.2	84.4	84.4	83.9	82.8	**84.6**
CR	88.9	88.9	88.2	88.3	88.5	87.1	**89.4**

Table 5.8: Comparing our random templates/demonstrations as data augmentation to SR, RI, RS, RD and EDA. Numbers are average of 5 train-test splits.

The results in Table 5.8 are obtained by applying SR, RI, RS, RD, EDA for 10% of input tokens. In contrast to ours, EDA, etc., for SupCon lead to worse performances than the baseline method in many tasks.

Ablations: Variable templates, demonstrations So far, we have shown the results by our method generating multi-views of inputs by appending *both* random templates and demonstrations. However, we find that in some tasks fixed templates with random demonstrations or random templates with fixed demonstration lead to even stronger performances (see Table 5.9). For example, sampling demonstrations with fixed templates for MRPC achieves a very strong result (80.0), outperforming all other methods in Table 5.8.

Task	LM-BFF	- demo + temp	+ demo - temp	+ demo + temp
SST-2 (acc)	89.2 (1.3)	**90.8** (0.3)	90.5 (0.4)	90.6 (0.1)
Subj (acc)	88.6 (3.3)	**90.8** (0.8)	90.6 (1.2)	90.4 (1.1)
SST-5 (acc)	47.9 (0.8)	49.3 (1.7)	48.9 (1.8)	**49.5** (1.1)
CoLA (Matt.)	6.1 (5.3)	9.9 (7.5)	8.5 (5.6)	**10.2** (5.8)
TREC (acc)	82.8 (3.1)	83.4 (0.5)	**86.7** (1.0)	83.3 (1.5)
MNLI (acc)	61.0 (2.1)	63.4 (3.3)	63.0 (3.2)	**64.0** (2.0)
MNLI-mm (acc)	62.5 (2.1)	65.5 (3.1)	64.9 (3.4)	**65.5** (2.7)
SNLI (acc)	66.9 (2.4)	69.8 (2.4)	68.5 (1.9)	**69.9** (2.4)
QNLI (acc)	60.7 (1.7)	65.4 (3.1)	**67.0** (3.6)	66.4 (3.5)
QQP (acc)	62.5 (2.6)	**68.9** (3.2)	67.8 (1.4)	68.8 (3.8)
RTE (acc)	64.3 (2.7)	64.9 (3.8)	62.6 (2.8)	**65.1** (3.5)
MRPC (F1)	75.5 (5.2)	79.0 (1.8)	**80.0** (1.8)	78.2 (3.1)
MR (acc)	83.3 (1.4)	85.8 (0.7)	85.4 (0.3)	**85.8** (0.6)
MPQA (acc)	83.6 (1.8)	84.0 (1.9)	84.1 (2.0)	**84.6** (1.5)
CR (acc)	88.9 (1.0)	88.6 (0.6)	88.2 (1.0)	**89.4** (1.0)

Table 5.9: Different strategies to construct multi-views of input sentences. Fixed demonstrations and sampling templates (- demo + temp), sampling demonstrations and fixed templates (+ demo - temp) and sampling both demonstrations and templates (+ demo + temp).

5.3.3. Summary

Limitations: Since SupCon clusters examples on class level, our framework applies only to classification tasks. Also, our framework requires large GPU memory, as SupCon is an in-batch contrastive loss that needs a large batch size.

Conclusion: We proposed a novel supervised contrastive learning framework and an effective augmentation method using prompts that can boost the performance of prompt-based language learners and outperform recent work on 15 few-shot tasks.

Our effective fine-tuning methods for Language Models also provide insights into VLM fine-tuning. Given that many VLMs employ frozen LLMs as language decoders conditioned on soft-prompts, our approach can be readily adapted to VLM fine-tuning with minimal modifications.

Chapter 6

LMMs in Bionformatics

Section 6.1

T-Cell Receptor-Peptide Interaction Prediction

In this section, we present "T-Cell Receptor-Peptide Interaction Prediction with Physical Model Augmented Pseudo-Labeling" [83]. The work has been published in KDD 2022.

6.1.1. Overview

Predicting the interactions between T-cell receptors (TCRs) and peptides is crucial for the development of personalized medicine and targeted vaccine in immunotherapy. Current datasets for training deep learning models of this purpose remain constrained without diverse TCRs and peptides. To combat the data scarcity issue presented in the current datasets, we propose to extend the training dataset by physical modeling of TCR-peptide pairs. Specifically, we compute the docking energies between auxiliary unknown TCR-peptide pairs as surrogate training labels. Then, we use these extended example-label pairs to train our model in a supervised fashion. Finally, we find that the AUC score for the prediction of the model can be further improved by pseudo-labeling

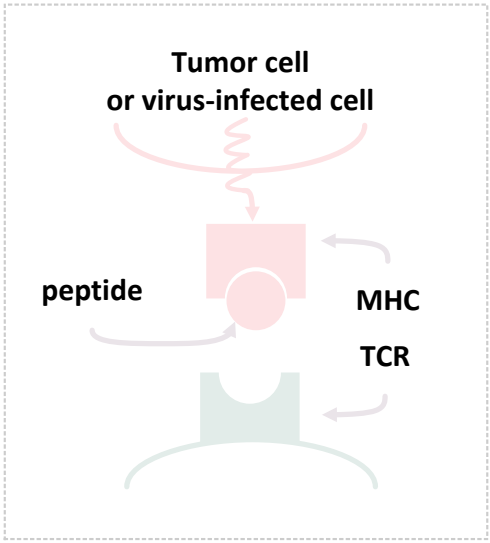

Figure 6.1: Illustration of T-cell receptors (TCR) and peptide binding: The TCR lies on the surface of the T-cell for recognition of foreign peptides. Peptides are presented by major histocompatibility complex (MHC) found on the surface of tumor cells or virus-infected cells. Common datasets for studying TCR-peptide interactions contain sequences of peptides and sequences of β chain of CDR3 of TCRs.

of such unknown TCR-peptide pairs (by a trained teacher model), and re-training the model with those pseudo-labeled TCR-peptide pairs. Our proposed method that trains the deep neural network with physical modeling and data-augmented pseudo-labeling improves over baselines in the available two datasets. We also introduce a new dataset that contains over 80,000 unknown TCR-peptide pairs with docking energy scores.

6.1.2. Motivation

T cells play an important role in the human immune response system by recognizing anomalous peptides through T-cell receptors (TCRs), which are protein complexes on the surface of T cells [34, 103]. Foreign peptides are presented by major histocompatibility complex (MHC) of tumor cells or virus-infected cells (shown in Figure 6.1). The binding between the TCR and peptide-MHC triggers further immune responses [61]. Thus, successfully predicting the interactions between TCRs and peptides is a key

step for the development of personalized medicine and vaccines, which is called the holy grail of immunology [28].

The TCR is a dimer with two chains: an α chain and a β chain. Each chain has three loops as complementarity-determining regions (CDR), denoted as CDR1, CDR2 and CDR3. CDR1 and CDR2 are primarily responsible for interactions with MHC and CDR3 interacts with peptides [166]. It is believed that the β chain of CDR3 has higher variations and is mainly responsible for the recognition of different peptides [105]. Thus, commonly widely used datasets (e.g., VDJdb [179] and McPAS [197]) for studying TCR-peptide interactions contain mainly sequences of β chain of CDR3 of TCRs and sequences of peptides.

Following the recent advancement of deep learning, several computational methods [88, 92, 137, 186, 187, 200, 212] for predictions of TCR-peptide interactions have been proposed. However, these methods mostly rely on the available labeled TCR-peptide pairs, despite that there are large public available TCR (without known associated peptides) sequences presented in the database.

In this study, to fully leverage the computational capability of a powerful neural network, we propose to learn our model with the external unlabeled TCRs in two ways: (1) data-augmented pseudo-labeling of TCR-peptide pairs by a model first trained on the labeled dataset (then re-train the model), and (2) physical modeling between TCRs and peptides by docking [222]. We find in experiments that these two approaches effectively improve the performance of models in two widely studied datasets.

6.1.3. Related Work

Conventional methods for predicting TCR-peptide interactions include nearest neighbor (SwarmTCR [44]), distance-based minimization (TCRdist [32]), PCA with decision tree [198], and Random Forest [35, 58].

Recently, several deep learning approaches have been proposed to predict the interactions between TCRs and peptides. ERGO [186] uses dual encoders for sequences of TCRs and peptides to predict the interactions based on β chain of CDR3. The second version of ERGO [187] further takes other information into considerations (e.g., α chain of CDR3, V and J gene, MHC type, T-cell type, etc.) to improve the predictions. TCRGP [88] makes predictions by Gaussian Process for some certain epitopes. NetTCR 1.0 [92] uses stacked convolutional neural network (CNN) for TCR-peptide predictions and further, NetTCR 2.0 [137] considers both α and β chain of CDR3. Besides the LSTM and CNN based models used in the aforementioned methods, TITAN [212] studies this problem by an attention-based [200] network.

Our method is also based on a deep learning approach. However, instead of focusing on designing the architecture of the model, we emphasize on computing the physical properties of TCR-peptide pairs (by leveraging a large available TCR database without known associated peptides) to extend the training dataset. Our method applies to any deep learning approaches that encode TCR and peptide sequences for predictions. The framework is also generalizable to study other protein-protein interactions.

Physical Modeling by Docking Docking [65, 151, 168] is a computational method for predicting the structures of protein complex (e.g., dimer of two molecules) given the structure of each monomer. It searches the configuration of the complex by minimizing an energy scoring function. In this study, we use the final docking energy (of the optimal structure of the complex) between a TCR and peptide as the surrogate binding label for the TCR-peptide pair.

Specifically in this study, we use HDOCK [222] as our docking algorithm. For a TCR/peptide sequence without known structure, HDOCK first uses a fast protein sequence searching algorithm [148, 160] to find the multiple-sequence-alignment (MSA) of the target sequence, and the corresponding structures in Protein Data Bank (PDB)

[10]. Then, it predicts the structures of the target sequence from the MSA and the known structures of homologous sequences. Finally, HDOCK optimizes the energy scores between the TCR and peptide, based on the predicted structures.

Our learning algorithm leverages the final docking energy score as a surrogate label for a TCR-peptide pair. We use a threshold to partition TCR-peptide pairs into negative pairs, positive pairs, and others.

Pseudo-labeling and Self-training Pseudo-labeling or self-training corresponds to first learning a model (teacher model) on the labeled dataset, and use the learned model (teacher model) to pseudo-label the unlabeled dataset. Finally, a new model is learned from the joint dataset of original labeled dataset and the extended pseudo-labeled dataset. Pseudo-labeling is a well-established method in semi-supervised learning including image classification [77, 149, 164, 184, 234], semantic segmentation [22, 66, 76, 144, 240], and many language tasks [6, 67, 81, 203].

Fixmatch [184] learns from unlabeled examples by matching the predictions of the model on weakly-augmented examples and heavily-augmented examples, which has impressive performances in several semi-supervised benchmarks.

Pham et al. [149] learn pseudo-labels by gradient-based meta-learning, i.e., the pseudo-labels are optimized for minimizing the validation loss of the target task. This Meta-Pseudo-Label approach achieves the best top-1 accuracy on ImageNet benchmark by leveraging a private weakly-labeled dataset with over 300M images.

Our method can be viewed as a semi-supervised problem by using a large database of TCR sequences without known associated peptides. Our study combines two approaches for assigning pseudo-scores to unknown pairs, i.e., one by a teacher model which is similar to Pham et al. [149], Rizve et al. [164], Sohn et al. [184], Zhang et al. [234], and another approach by assigning pseudo-labels from properties of the physical modeling of the TCR-peptide pairs.

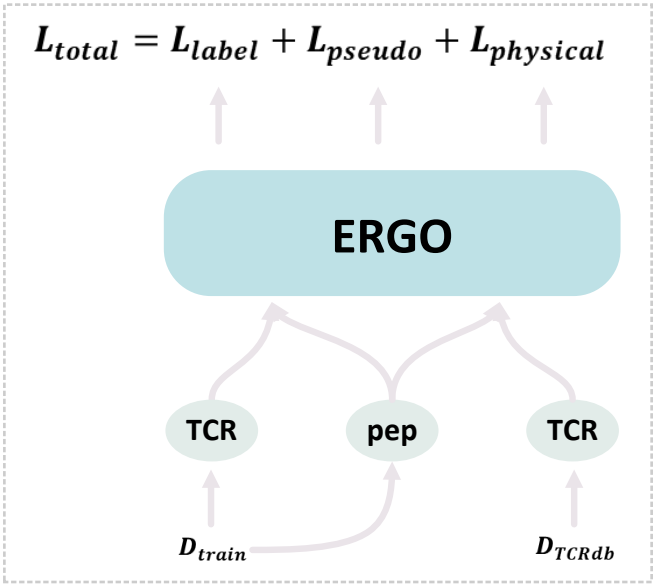

Figure 6.2: Overview of our method. Our method learns a TCR-peptide interaction model (based on ERGO) by three losses: a standard cross-entropy loss from examples of labeled dataset, a KL-divergence loss from pseudo-labeled examples (by a teacher model), and finally a cross-entropy loss based on the physical properties (i.e., docking energies) between TCRs and peptides.

6.1.4. Methods

Problem Setup. Let us denote t for a TCR sequence, p for a peptide sequence, and $x = (t, p)$ is a TCR-peptide pair. We have a TCR-peptide dataset $\mathcal{D} : \{(x_i, y_i)\}$ where $i = 1, 2, ..., n$ and n is the size of the dataset $\mathcal{D}$. x_i represents a TCR-peptide pair and y_i is either 1 indicating a positive pair, or 0 indicating a negative pair. Our goal is to learn a model from $\mathcal{D}_{\text{train}}$ that performs well on the testing dataset $\mathcal{D}_{\text{test}}$, where $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ are a split of dataset $\mathcal{D}$.

The data scarcity issue presented in $\mathcal{D}_{\text{train}}$ limits the model's generalization on $\mathcal{D}_{\text{test}}$. Thus, to improve the performance, we leverage a TCR database which has no associated peptides $\mathcal{D}_{\text{TCRdb}} : \{t_j\}$, where $j = 1, 2, ..., N$ and N is the number of TCRs in $\mathcal{D}_{\text{TCRdb}}$. The number of TCRs in $\mathcal{D}_{\text{TCRdb}}$ is much larger than the number of TCRs in $\mathcal{D}_{\text{train}}$, i.e., $N \gg n$. Note that the TCR in $\mathcal{D}_{\text{TCRdb}}$ has no known interaction with

peptides in $\mathcal{D}_{\text{train}}$.

Overview. Our method trains a deep learning model for predicting TCR-peptide interactions from 3 losses: a supervised cross-entropy loss from the given known TCR-peptide pairs (illustrated in Section 6.1.4), a supervised cross-entropy loss based on docking energies of unknown TCR-peptide pairs (illustrated in Section 6.1.4), and a KL-divergence loss from the pseudo-labeled (by a teacher model) unknown TCR-peptide pairs (illustrated in Section 6.1.4).

Base Model. In our study, we use ERGO-I [186] as our base models for all experiments. ERGO-II [187] improves over ERGO-I by further considering auxiliary information, i.e., α chain of CDR3, V and J gene, MHC types and T-cell types. We choose ERGO-I over ERGO-II for the following reasons: Our goal is to show that a machine learning model for predicting the interaction of two molecules can be improved by further physical modeling between them. ERGO-I is a general framework that can be adapted to study any protein-protein interactions. Whereas, ERGO-II is only applicable to TCR-peptide interaction predictions. We expect that our framework to work beyond the TCR-peptide predictions, though our experiments are focused on this specific interaction.

Learning from Known pairs. ERGO [186] has two separate encoders: $f_{\theta_{\text{TCR}}}$ and $f_{\theta_{\text{pep}}}$ for TCRs and peptides respectively. The encoder for TCRs is a stacked MLPs and pre-trained by an auto-encoding loss, whereas the encoder for peptides is parameterized by a LSTM [71] (In another variant of ERGO, both encoders for TCRs and peptides are LSTMs, see Springer et al. [186] for more details). For a pair

$x = (t, p) \in \mathcal{D}_{\text{train}}$, the embedding of TCR and peptide can be computed by

$$e_{\text{TCR}} = f_{\theta_{\text{TCR}}}(t) \tag{6.1}$$

$$e_{\text{peptide}} = f_{\theta_{\text{pep}}}(p) \tag{6.2}$$

A fully connected MLP $f_{\theta_{\text{clf}}}$ is then attached to the concatenated embeddings of TCRs and peptides to perform the final classification:

$$pred = f_{\theta_{\text{clf}}}(\text{concat}(e_{\text{TCR}}, e_{\text{peptide}})) \tag{6.3}$$

For simplicity, in the following part of the section, we will denote

$$pred = f_{\Theta}(t, p) \tag{6.4}$$

$$= f_{\theta_{\text{clf}}, \theta_{\text{TCR}}, \theta_{\text{pep}}}(t, p) \tag{6.5}$$

where f_{Θ} is the full model that contains $f_{\theta_{\text{clf}}}, f_{\theta_{\text{TCR}}}, f_{\theta_{\text{pep}}}$. The final classification loss is the binary-cross-entropy between the prediction $pred$ and the label y for this TCR-peptide pair $x = (t, p)$.

$$\mathcal{L}_{labeled} = \text{BinaryCrossEntropy}(pred, y) \tag{6.6}$$

Learning from Physical Modeling. Due to lacking of diverse TCR and peptide pairs in the supervised training dataset $\mathcal{D}_{\text{train}}$, we propose to leverage the existing large amount of TCR sequences without associated peptides, by modeling the physical properties between these TCRs and peptides (from the training set $\mathcal{D}_{\text{train}}$) to extend our training dataset $\mathcal{D}_{\text{train}}$.

More accurate physical modeling can be achieved by running molecular dynamic

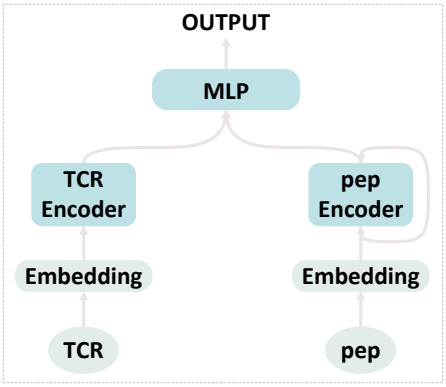

Figure 6.3: Models used in our study. The model is borrowed from ERGO [186] for fair comparisons. The model has two separate encoders for TCR and peptide. Following Springer et al. [186], we experiment with both LSTM and AE model for TCR encoder and only LSTM for peptides. A MLP is attached on top of concatenated representation of the TCR and peptide to perform the final classification. The classification loss is a Binary Cross Entropy (BCE) loss.

(MD) simulations which are computationally heavy (It can take 20 hours for a single TCR-peptide pair [4]). We choose to use the docking energy between a TCR and peptide as an indication of interaction, due to its simplicity so that we can apply it to large-scale unlabeled TCRs (i.e., computing docking energy takes around 2 minutes for each TCR-peptide pair, running on Intel I7-7700K with 64GB RAM). Docking energy reflects the binding affinity between molecules by treating molecules as rigid bodies [145]. Docking of a peptide onto a TCR finds the optimal configuration of two rigid bodies with the minimal energy by moving the peptide around the surface of the TCR. Thus, the smaller docking energy indicates a likely positive pair of the given TCR and peptide.

Docking is a physics-based modeling that first requires the known structures of TCRs and peptides. Given a TCR sequence t' sample from $\mathcal{D}_{\text{TCRdb}}$, and a peptide sequence p' from $\mathcal{D}_{\text{train}}$, we first build structures of the TCR t' and the peptide p' by using blastp [17] to find homologous sequences with known structures. Then, we call MODELLER [211] for building structures for TCRs and peptides. Once we

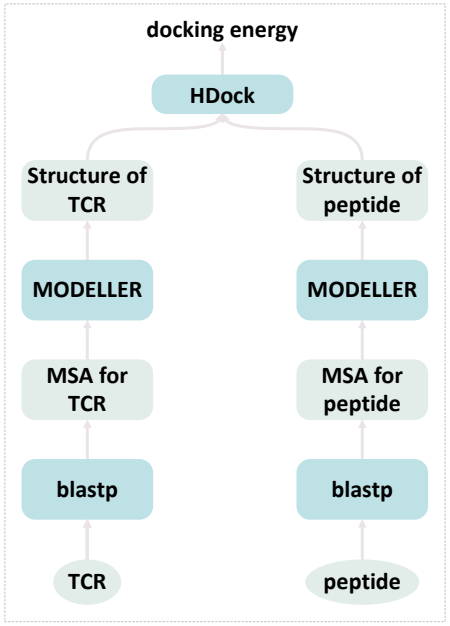

Figure 6.4: Overview of docking using HDOCK. For a given sequence of TCR/peptide, we first use blastp to find the multiple-sequence alignment (MSA) for the sequence. MSA and the corresponding structures from PDB are then used by MODELLER for building the structures of the TCR/peptide. Finally, we call HDOCK with the given structures of the TCR and peptide for computing docking energies.

have structures of TCRs and peptides, we use HDOCK [222] for docking TCRs and peptides. In this way, we build 80K TCR-peptide pairs with docking energy scores (see also Figure 6.4). We then pseudo-label these pairs with the bottom 25% energy scores to be positive pairs and those with top 25% energy scores to be negative pairs. Thus, we end up with a dataset pseudo-labeled by docking energies: $\mathcal{D}_{\text{auxiliary}}$. For $((t', p'), y') \in \mathcal{D}_{\text{auxiliary}}$, where y' is the pseudo-label by docking, the learning objective is then:

$$pred' = f_\Theta(t', p') \tag{6.7}$$

$$\mathcal{L}_{\text{physical}} = \text{BinaryCrossEntropy}(pred', y') \tag{6.8}$$

Learning from Pseudo-labeled Pairs. The introduction of $\mathcal{D}_{\text{auxiliary}}$ makes our learning problem equivalently into a semi-supervised setting. Besides the pseudo-labeling by physical modeling described in Section 6.1.4, we can also leverage well-established semi-supervised methods to further improve the results. Pseudo-labeling by a teacher model is proven to be a successful technique in semi-supervised learning [185]. The algorithm first labels unlabeled examples with a model (teacher model) first trained on the labeled dataset. Then it re-trains the model with labeled training dataset, combining with the extended pseudo-labeled examples (see also Figure 6.5).

Following Section 6.1.4, first training with $\mathcal{D}_{\text{train}}$ with only loss $\mathcal{L}_{\text{label}}$ leads to a model Θ_{teacher}. For a pair (t', p') sampled from $\mathcal{D}_{\text{auxiliary}}$,

$$prob' = f_{\Theta_{\text{teacher}}}(t', p') \tag{6.9}$$

where $prob'$ is the output probability of the teacher model that we use as the pseudo-label for this TCR-peptide pair (t', p'). The learning objective function for the pseudo-labeled examples by the teacher model is then:

$$pred' = f_{\Theta}(t', p') \tag{6.10}$$

$$\mathcal{L}_{\text{pseudo-labeled}} = \text{KL-div}(pred', prob') \tag{6.11}$$

Here, we use KL-divergence (KL-div) for matching the model's predictions to the teacher's output probabilities. The final total training loss is the combination of the three losses:

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{\text{labeled}} + \beta\mathcal{L}_{\text{physical}} + \gamma\mathcal{L}_{\text{pseudo-labeled}} \tag{6.12}$$

In our experiments, we set α, β, γ to be 1. However, we expect better performances

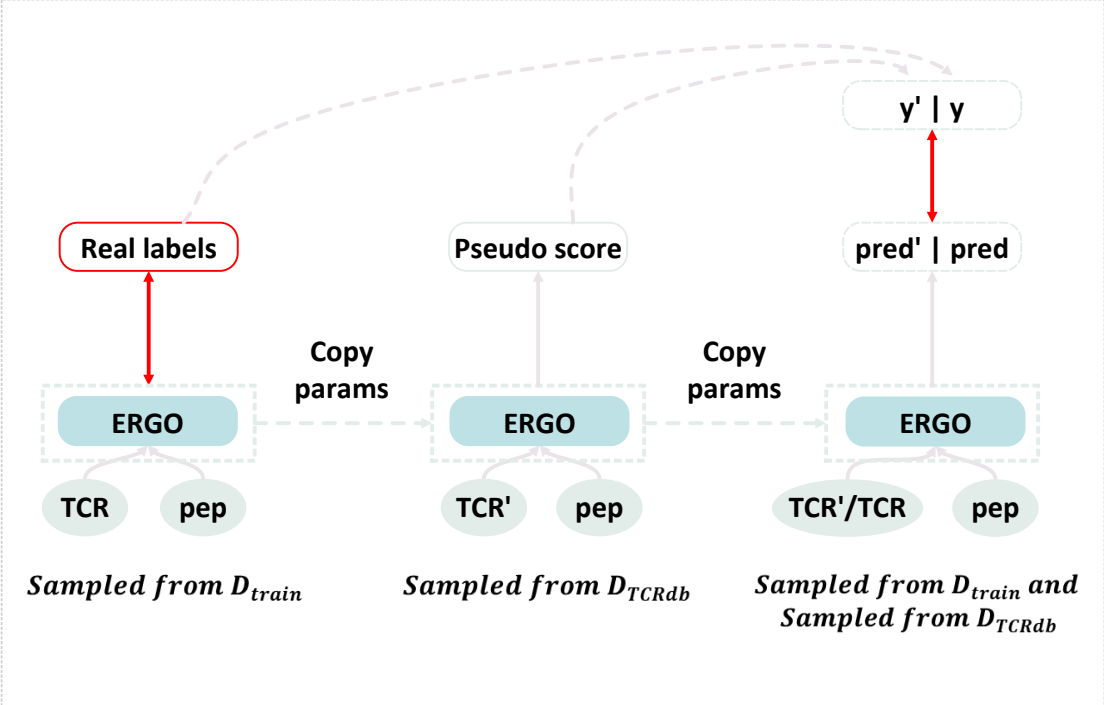

Figure 6.5: Overview of learning from data-augmented pseudo-labeling. An ERGO model is first learned with TCRs and peptides sample from $\mathcal{D}_{\text{train}}$, and this model is used as the teacher model. Then, this teacher model is used for pseudo-labeling TCR-peptide pairs from auxiliary dataset. Finally, we re-train an ERGO model with the original dataset and the extended pseudo-labeled dataset.

by further tuning these hyper-parameters.

Look Ahead Meta-update. While learning from physical modeling effectively extends the training dataset, the success of the learning also relies on the quality of the physical modeling. We want to learn the model such that the auxiliary learning from the physical modeling is optimized for the primary learning objective (the loss on the test set). This is usually done by meta-learning that minimizes a validation loss. The meta-learning algorithm introduces a gradient-on-gradient learning procedure that is time-consuming [50]. Thus, we borrow the idea of meta-learning, but instead of minimizing a validation loss, we approximate it with minimizing the training loss of current batch, i.e., we optimize the gradients from learning with physical modeling

such that gradients from this auxiliary objective only reduce the training loss ($\mathcal{L}_{\text{label}}$) on the current batch.

Look Ahead Meta-update: For each training iteration, we first sample a batch (x, y) from $\mathcal{D}_{\text{train}}$, and a batch (x', y') from $\mathcal{D}_{\text{auxiliary}}$. Then, we compute the loss $\mathcal{L}_{\text{labeled}}$ using (x, y) (see details in Section 6.1.4) and $\mathcal{L}_{\text{pseudo-labeled}}$ using x' and a teacher model (described in Section 6.1.4). Next, we update the parameters of the model using gradients from their two losses accordingly and denote the parameters as Θ_{t-1}. Lastly, we compute the loss $\mathcal{L}_{\text{physical}}$ using (x', y') (described in Section 6.1.4) and update the model one step further to be Θ_t. if

$$\text{CrossEntropy}(f_{\Theta_{t-1}}(x), y) \leq \text{CrossEntropy}(f_{\Theta_t}(x), y) \tag{6.13}$$

i.e., learning the current batch with physical modeling leads to larger training error, we then switch the model back to Θ_{t-1}.

$$\Theta_t \leftarrow \Theta_{t-1} \tag{6.14}$$

In other words, we do not update the parameters of the model if gradients of learning from physical modeling do not help the training process. To compensate the reducing number of training examples, we double the learning rate for $\mathcal{L}_{\text{physical}}$ when applying this meta-update.

6.1.5. Experiments

Datasets. We evaluate our method on two datasets, i.e., McPAS [197] and VDJdb [179]. McPAS is a manually curated dataset with more than 20,000 TCRs with matching over 300 peptides. Similarly, VDJdb dataset has over 40,000 TCRs paring with around 200 peptides. We follow the TCR-Peptide Paring studied in Springer et al.

[186] and split the dataset into 80% for training and 20% for testing, i.e., $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$. Because McPAS and VDJdb have only known positive pairs between TCRs and peptides, we follow Springer et al. [186] that samples a random TCR and a random peptide to form a negative pair, and samples 4× more negative pairs than the positive pairs. In experiments, we investigate with different sizes of $\mathcal{D}_{\text{train}}$, i.e, 6K, 10K and 20K. Our $\mathcal{D}_{\text{TCRdb}}$ has TCRs from benny chain TCR memory data of Springer et al. [186] (which was only used for unsupervised pre-training of a TCR autoencoder model). We compute the docking energies for 80K pairs and thus $|\mathcal{D}_{\text{auxiliary}}| = 40K$ (Because we only use pairs with top and bottom 25% energy scores, see details in Section 6.1.4).

We will make the original $\mathcal{D}_{\text{auxiliary (full)}}$ which has computed docking scores for over 80,000 TCR-peptide pairs available. [1]

Models and Training Details. We use the same model architecture (ERGO, see also Figure 6.3) from Springer et al. [186]. ERGO has two encoders: one for TCRs which encodes the one-hot representation of a TCR sequence with amino acid embeddings of dimension 10, followed by MLPs with hidden sizes $300, 100, 30$. The other encoder for peptides is parameterized by a LSTM, which has two layers, each with dimension of 100. The last hidden states of LSTM are used as the representation for peptides. We denote this model as **AE-LSTM** model.

All models are trained with Adam optimizer [99] with fixed learning rate of $5e^{-4}$, batch size of 50, epochs of 100, and with 5 different random seeds.

We also experiment with another variant of ERGO that has both LSTM encoders for TCRs and peptides. We denote this model as **double-LSTM** model. The two LSTMs are symmetric, each with two layers and hidden dimension of 100.

[1] https://github.com/yiren-jian/Tcell-Peptide-PhyAugmentation

Results on McPAS. We show in Table 6.1 and 6.2 for investigating McPAS with 2 different variants of ERGO (following Springer et al. [186]), i.e., one with AE encoder for TCRs and one with LSTM for TCRs. Both models have the same LSTM encoder for peptides.

AE-LSTM model: We see that in Table 6.1 data-augmented pseudo-labeling improves the AUC score by 4.1 and 6.4 with $|\mathcal{D}_{\text{train}}|$ of 6K and 10K. Docking (physical modeling) further increases the AUC score by 2.9 and 2.1. The overall better results by docking is likely due to the fact the physical properties of molecule complex introduces learning signals from another modality, whereas data-augmented pseudo-labeling still relies on the teacher model which is only trained on the original dataset $\mathcal{D}_{\text{train}}$. The overall better performances by physical modeling emphasizes the better pseudo-labels based on docking, comparing to a teacher model learned from original limited training set. We also notice that the improvements reduce when the training dataset grows larger.

These improved performances emphasize the importance of having more diverse TCR-peptide pairs during the training with data-augmented pseudo-labeling and physical modeling. The meta-update introduced in Section 6.1.4 updates the parameters of the model by only learning from useful signals from physical modeling, that achieves the best results by increasing an average of 0.9 over the vanilla sum of the three losses.

double-LSTM model: Our method generalizes to different base models. Here we show that our framework is able to improve over the baselines using a double-LSTM as ERGO base models in Table 6.2. In this case, data-augmented pseudo-labeling works about equally well to physical modeling. The best models of our method outperform vanilla ERGO by 3.9, 2.8 and 1.8 with $|\mathcal{D}_{\text{train}}|$ of 6K, 10K and 20K.

Results on VDJdb. **AE-LSTM** model: In Table 6.2, we see a similar trend in VDJdb dataset. ERGO + docking improves over the baseline by 1.5, 3.6 and 4.7 with

Data size	6K	10K	20K
ERGO	54.4 ± 0.5	56.3 ± 0.5	71.2± 0.3
+ Pseudo	58.5 ± 0.5	62.7 ± 0.4	72.7± 0.3
+ Docking	61.4 ± 0.4	64.8 ± 0.4	72.4± 0.4
ours (3 losses)	62.1 ± 0.4	66.0 ± 0.4	73.2± 0.3
ours + meta-update	**63.4± 0.4**	**66.5± 0.4**	**74.2± 0.3**

Table 6.1: Experimental results on McPAS using base model of ERGO-AE. ERGO: Baseline method, ERGO + Pseudo: ERGO with data-augmented pseudo-labeling, ERGO + Docking: ERGO with physical modeling, ours (3 losses): ERGO with data-augmented pseudo-labeling and physical modeling, ours+ meta-update: ours (3 losses) with meta-update described in Section 6.1.4. Data size denotes the different sizes of $\mathcal{D}_{\text{train}}$. Results are collected from 5 different independent experimental runs.

Data size	6K	10K	20K
ERGO	67.6 ± 0.4	71.9 ± 0.4	76.6 ± 0.3
+ Pseudo	69.3 ± 0.4	73.6 ± 0.3	77.6 ± 0.3
+ Docking	69.4 ± 0.4	73.3 ± 0.3	77.9 ± 0.2
ours (3 losses)	70.4 ± 0.3	73.7 ± 0.3	77.6 ± 0.2
ours + meta-update	**71.5 ± 0.3**	**74.7 ± 0.3**	**78.4±0.2**

Table 6.2: Experimental results on McPAS using base model of ERGO-LSTM. Results are collected from 5 different independent experimental runs. In these experiments, ERGO+Psudo and ERGO+Docking perform roughly equally well.

Data size	6K	10K	20K
ERGO	60.7 ± 0.5	61.0 ± 0.5	66.8 ± 0.4
+ Pseudo	61.0 ± 0.5	63.9 ± 0.4	69.8± 0.3
+ Docking	62.2 ± 0.5	64.6 ± 0.5	71.5 ± 0.3
ours (3 losses)	63.4 ± 0.5	66.4 ± 0.4	72.2 ± 0.3
ours + meta-update	**64.6 ± 0.5**	**67.6 ± 0.4**	**72.9 ± 0.3**

Table 6.3: Experimental results on VDJdb using base model of ERGO-AE. Results are collected from 5 different independent experimental runs.

$|\mathcal{D}_{\text{train}}|$ of 6K, 10K and 20K. Our method with three losses achieves best results in all 3 tasks.

double-LSTM model: In Table 6.4, we find that data-augmented pseudo-labeling only outperforms the baseline marginally in 3 tasks (0.3, 0.4, and 0.3). This is possibly

Data size	6K	10K	20K
ERGO	68.1± 0.4	72.0 ± 0.3	73.6 ± 0.4
+ Pseudo	68.4 ± 0.3	72.4 ± 0.3	73.9 ± 0.3
+ Docking	69.5 ± 0.4	73.4± 0.3	74.6± 0.3
ours (3 losses)	70.4± 0.3	72.9± 0.3	74.6± 0.3
ours + meta-update	**71.5 ± 0.3**	**73.8± 0.3**	**75.2± 0.3**

Table 6.4: Experimental results on VDJdb using base model of ERGO-LSTM. Results are collected from 5 different independent experimental runs. In these experiments, ERGO+Pseudo only improves over the baseline marginally, while physical modeling by docking still increase the AUC by significant margins.

due to the fact that the teacher model by double-LSTM model fails to generate useful pseudo-labels for extended TCRs for re-training the model. However, physical modeling by docking consistently improves over the baseline by considerable margins in all 3 tasks.

The results indicate that while data-augmentation may fail sometimes, physical modeling can always provide the pseudo-labels for unlabeled TCRs for effective training.

6.1.6. Discussion

There's a long tail distribution of peptides presented in McPAS and VDJdb datasets. For example in McPAS, peptide LPRRSGAAGA has over 2000 known TCR pairs while many others are only paired with less than 10 TCRs. The baseline models learning directly from such unbalanced dataset fails to generalize well on the testing set for those rare peptides. For example using AE-LSTM model with 6K labeled training examples in McPAS, we find that baseline method for prediction of a rare peptide KRWIILGLNK has only AUC score of 52.8, while our method achieves 68.1. Note that the average AUC for all peptides is 54.4. The results indicate that our method significantly improves the results for rare peptides, where the baseline is struggling.

We show more examples in Table 6.5.

rare peptides	baseline	average	ours
KRWIILGLNK	52.8	54.4	68.1
KMVAVFYTT	48.9	54.4	65.8
FPRPWLHGL	50.2	54.4	58.5

Table 6.5: Experiments with AE-LSTM model with McPAS dataset of 6K labeled examples (from $\mathcal{D}_{\text{train}}$). "average" denotes the average AUC for all peptides in this experimental setup.

6.1.7. Conclusion

In this work, we investigate several techniques to improve the prediction of TCR-peptide interactions. Specifically, we find that pseudo-labeling of unknown TCR-peptide pairs from auxiliary dataset and re-training the model with the mixture of original dataset and extended pseudo-labeled dataset can improve the results. Further, docking energies as the physical properties between TCR-peptide pairs can be used as surrogate pseudo-labels for training the deep learning model. And pseudo-labels by physical modeling is generally better than pseudo-labels by a teacher model trained from the original training set. At last, we propose a meta-update technique that further updates the parameters of the model by selecting only positive gradients of learning from physical modeling. Experiments on two widely studied datasets demonstrate the effectiveness of our proposed approaches.

Section 6.2

Learning RNA tasks from Protein LLMs

In this section, we present “Knowledge from Large-Scale Protein Contact Prediction Models Can be Transferred to the Data-Scarce RNA Contact Prediction Task” [85]. The work is published in arXiv pre-print server in 2023.

6.2.1. Overview

RNA, whose functionality is largely determined by its structure, plays an important role in many biological activities. The prediction of pairwise structural proximity between each nucleotide of an RNA sequence can characterize the structural information of the RNA. Historically, this problem has been tackled by machine learning models using expert-engineered features and trained on scarce labeled datasets. Here, we find that the knowledge learned by a protein-coevolution Transformer-based language model can be transferred to the RNA contact prediction task. As protein datasets are orders of magnitude larger than those for RNA contact prediction, our findings and the subsequent framework greatly reduce the data scarcity bottleneck. Experiments confirm that RNA contact prediction through transfer learning using a publicly available protein language-model is greatly improved. Our findings indicate that the learned structural patterns of proteins can be transferred to RNAs, opening up potential new avenues for research.

6.2.2. Motivation

Proteins and RNAs are critical to many biological processes such as coding, regulation, and expression [46, 51, 63, 174, 177]. Understanding their structures is key to deciphering their functionalities. While experimental methods like X-ray diffraction [188], nuclear magnetic resonance (NMR) [19], and Cryogenic electron microscopy (Cryo-EM) [60] can determine 3D structures, it remains challenging for structurally flexible molecules, e.g., RNAs [133]. Consequently, the Protein Data Bank has limited RNA structures cataloged [10].

In response, many computational tools for 3D structure prediction of biological molecules have been developed in the last decade [93, 107, 223, 224]. Recently, deep neural networks such as AlphaFold [91], ProteinMPNN [33], RoseTTAFold [7], and Metagenomics [123] have revolutionized 3D protein structure prediction, partly due to

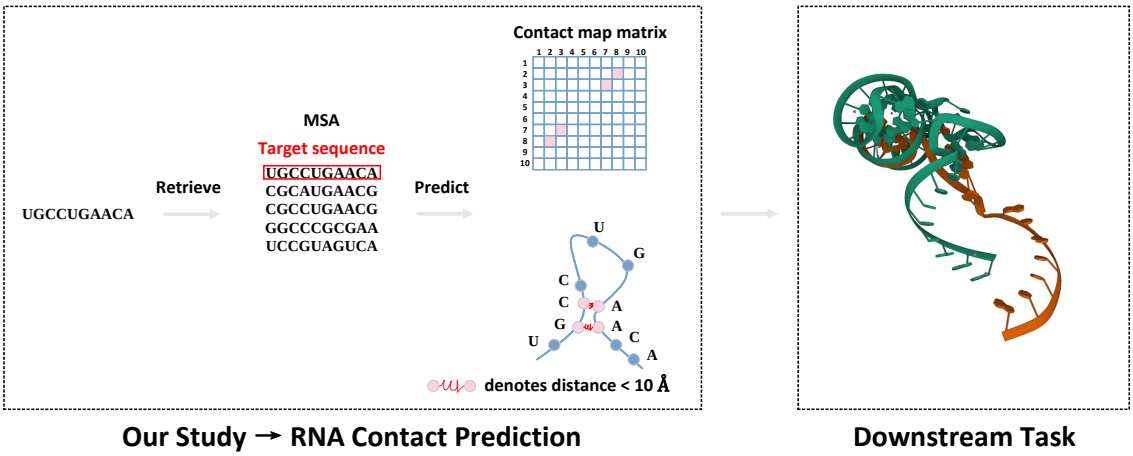

Figure 6.6: Our study is focused on RNA contact prediction, i.e., predicting the contact map matrix for an RNA sequence. The contact map indicates the proximity between each nucleotide, with those closer than a threshold (10 Å) being deemed in contact. Correct predictions of the contact map can benefit downstream tasks, e.g., by acting as constraints for filtering 3D RNA structure predictions.

their large size and training datasets. However, this progress has not been paralleled for RNAs, mainly due to the scarcity of RNA datasets. Current RNA datasets are significantly smaller than protein datasets, with well-curated datasets containing less than 100 RNAs [233] and models trained on fewer than 300 RNA structures [193, 243]. These small datasets are insufficient for training large deep neural networks, leading to RNA 3D prediction tools based on simulations (SimRNA, Rosetta FARFAR, iFoldRNA, NAST) [14, 31, 90, 104, 178] or fragment assembly (ModeRNA, Vfold, RNAComposer, 3dRNA) [153, 167, 205, 219, 246].

In the absence of powerful 3D structure prediction models, certain structural properties of RNAs can be determined through RNA Contact Prediction [78]. The contact predictions can be used as an intermediary step to facilitate the prediction of 3D structures or directly for downstream tasks that rely on RNA structural information. For an RNA sequence of length L, this task aims at predicting a $L \times L$ symmetric binary matrix (called contact map) where a value of 1 at position (i, j) indicates that

the i^{th} and j^{th} nucleotides are in contact[2] with each other to each other in 3D space. The predicted contact maps capture structural constraints, which can be used for downstream tasks, such as refining RNA 3D prediction tools [205] (see Figure 6.6 for an overview of the task). Note that for a target RNA sequence, the input for an RNA contact prediction model is an RNA multiple sequence alignment (MSA), which corresponds to the target RNA sequence stacked with known homologous sequences.

The first RNA Contact Prediction attempt was Direct Coupling Analysis (DCA), with variants like mfDCA [138], mpDCA [215], plmDCA [45], bmDCA [140], and PSICOV [89]. These self-supervised methods infer contact maps using maximum likelihood estimation without labeled datasets. Recently, supervised methods have been explored to improve RNA contact prediction. Due to the small number of available RNA-contact map pairs, these methods rely on feature engineering, such as in RNAcontact [193], which uses covariance matrices from Infernal [141], PETfold-predicted secondary structures [171], and RNAsol solvent accessible surface areas [192] to train a deep ResNet model [69]. Zerihun et al. [233] found that simple DCA outputs re-weighted by a convolutional layer (CoCoNet) achieve comparable RNA contact prediction precision [233].

Supervised RNA contact prediction methods leverage additional knowledge from RNA analytic tools for more informative features. These small models are necessitated by limited training examples. In contrast, abundant protein data allowed for training a large Transformer-based deep neural network, Co-evolution Transformer (CoT), for protein contact prediction [237]. CoT was trained on 90K curated protein structures, compared to 0.1K RNA structures.

Since we lack the data to train such a model for RNA contact prediction from scratch, we investigate the possibility of re-using and tuning the learned parameters of a

[2]Contact is defined as distances smaller than a specific threshold. Following prior works, this is set by a hard distance threshold of 10 Å.

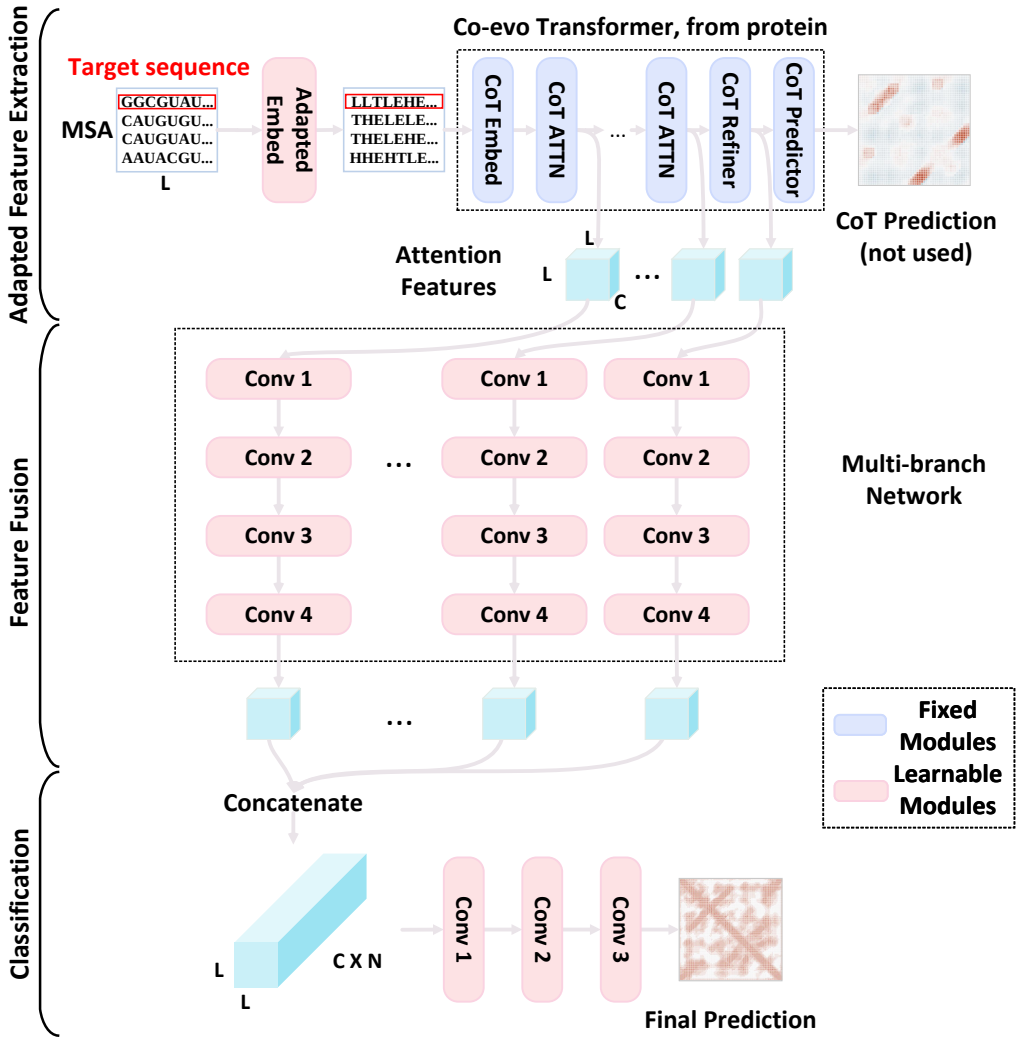

Figure 6.7: Overview of our three-stage method (from top to bottom). **Adapted Feature Extraction**: First, a projection layer is used to translate the RNA MSA sequences into protein language (e.g., from nucleotide "AUCG" to amino acids "HETL"). Then, we leverage a fixed large-scale pre-trained protein contact prediction transformer model (called Co-evolution Transformer model (CoT)) to extract attentive (i.e., contribution) features at different layers. **Feature Fusion**: Features from different layers are processed by separate convolution blocks before being concatenated. **Classification**: The aggregated features are sent into a standard Convolutional Network (ConvNet) classifier with three layers of convolution.

pre-trained protein language-model (such as CoT) to create an RNA contact prediction model, a process referred to as transfer learning. Inspired by recent breakthroughs in unified vision-language models [9, 113, 204] and transfer learning across text and visual

domains [79, 131], which have demonstrated the effectiveness of transferring knowledge between related modalities, such as leveraging the structural abilities learned from code and music to enhance language models [146], we propose that bio-molecule contact patterns learned by the CoT protein Transformer network could be transferred to improve RNA contact prediction performance.

Similar to RNA contact prediction models, CoT takes protein MSAs as input. The input to CoT is represented using English characters, with each amino acid represented by a unique English character. CoT then utilizes the attention mechanism of Transformers [200] to learn the contacts, analogous to how Transformer-based language models, such as GPT-3 [15], learn dependencies between words in a given text. Though at the surface level, RNA and protein sequence data are comprised of different building blocks (nucleotides for RNAs and amino acids for proteins), we speculate that they share deeper similarities concerning their contact patterns, analogous to two languages with different lexicons but a similar syntax. Hence, it may be possible to transfer knowledge about contact patterns from one to another, analogous to cross-lingual transfer in Transformer-based language models [62].

We investigate our hypothesis by adapting the pre-trained CoT to our RNA dataset and using the adapted representations to train a convolutional network (ConvNet) for RNA contact prediction (see Figure 6.7 for an overview of our method). Our explorations show that this simple method, which does not rely on any additional pre-processing or feature engineering and can detect true contacts missed by prior works. In addition to improving RNA contact prediction by using knowledge from a pre-trained protein language-model, our study serves as a strong proof of concept for the possibility of transfer learning between the proteins and RNAs.

6.2.3. Background and Related Works

Unsupervised Contact Prediction Based on the Co-Evolution Hypothesis

The *co-evolution hypothesis* is the basis of many contact prediction methods (for both proteins and RNA). The hypothesis suggests that spatially proximate pairs of amino acids or nucleotides tend to co-evolve to maintain their structure and function [215].

In practice, this is used for RNA (and protein) contact prediction as follows: to predict the contacts of a target RNA sequence, first, a sequence database is used to find similar sequences. These sequences are likely homologs of the target, with differences due to mutations during evolution. These sequences are then aligned, creating what is called a multiple sequence alignment (a.k.a MSA). Based on the co-evolution hypothesis, the contact prediction for the target sequence can then be reformulated as detecting the co-evolution nucleotide pairs in the MSA. For example, Morcos et al. [138] calculate the covariance between each pair of nucleotides, thus creating a covariance matrix as an approximation of the co-evolution. The direct coupling score (DCA score) between each pair can then be computed through different approximation methods. While Morcos et al. [138] use mean-field approximation (mfDCA), other DCA variants (e.g., [45]) use different tricks for the approximation.

DCA methods are all purely unsupervised, based on the counting frequency of the residues in MSA, and can be applied to proteins and RNA sequences. Recently, it has been shown that transformer-based protein language-models can also be unsupervised protein contact learners [156, 157], though these methods are not necessarily based on the co-evolution hypothesis.

Supervised Contact Prediction Given $\sim$ 180K known protein structures in PDB, Zhang et al. [237] train a 20M-parameters attention-based Transformer model for end-to-end prediction of protein contacts based on MSA. The attention mechanism of the model, called the Co-evolution Transformer (CoT), is specifically designed to

model co-evolution by considering the outer product of representations of two positions. Such a model is only successful given a large labeled dataset of known MSA to contact mappings.

Training such a large model for RNA is unfortunately not practical as there are currently no such large datasets available. The largest of such data is at least 2-3 orders of magnitude smaller than what is available for proteins. To overcome this bottleneck, most resort to feature engineering to train smaller models. For instance, recent works [193, 243] combine DCA outputs (or similarly, covariance matrices) with other features (such as predicted secondary structures, solvent surface areas, etc.) extracted from different RNA analysis tools to train relatively small convolution networks, using only hundreds of labeled data point. Finally, Zerihun et al. [233] propose CoCoNet, showing that the output of DCA by itself is sufficient for training such models and that oftentimes expensive additional feature extraction is not needed.

Transfer Learning We show in this chapter that the learned knowledge of a pre-trained protein contact prediction model can be effectively used for RNA contact prediction, not only removing the need for additional feature engineering and extraction but also vastly outperforming CoCoNet. Our proposed method is built upon the concept of "Transfer Learning" [39], which assumes that knowledge learned from one task is beneficial to other related tasks. Transfer learning has enabled the adaption of large pre-trained deep neural networks to new tasks with a limited number of labeled examples. This is typically done by training newly initialized layers at the end of the pre-trained network (which tends to be task-specific) using the small dataset while keeping the other layers frozen (which preserves the learned knowledge from the previous task). Only the relatively small set of parameters in the final layers will be updated, which will adapt the network to the new task.

Transfer Learning has been shown to be effective for class-level transfer in a single

domain [202] and for different domains (e.g., an image classification model adapted for semantic segmentation [129]). There is also research that shows that seemingly unrelated tasks can also help each other [135]. Even models for different modalities can be transferred. For instance, Mokady et al. [136] and Lu et al. [131] show that semantic knowledge can be transferred between language and visual models.

A key challenge of RNA contact prediction is the small dataset size, which prohibits us from learning a deep model from scratch. We hypothesize (and later verify) that knowledge could be effectively transferred from a pre-trained protein contact Transformer to RNA contact prediction, enabling us to train high-performing RNA contact prediction models without the need for additional labeled or feature engineering, both of which can be prohibitively expensive. Analogies can be drawn between our approach and research done on the cross-lingual transfer of language models [62] that adapt a pre-trained model to a new language by learning its syntax while retaining the semantic knowledge in the pre-trained model; here we are adapting a biological model pre-trained on the "language of proteins" to the "language of RNAs".

6.2.4. Backgrounds

Unsupervised Contact Prediction Based on the Co-Evolution Hypothesis

The *co-evolution hypothesis* is the basis of many contact prediction methods (for both proteins and RNA). The hypothesis suggests that spatially proximate pairs of amino acids or nucleotides tend to co-evolve to maintain their structure and function [215].

In practice, this is used for RNA (and protein) contact prediction as follows: to predict the contacts of a target RNA sequence, first, a sequence database is used to find similar sequences. These sequences are likely homologs of the target, with differences due to mutations during evolution. These sequences are then aligned, creating what is called a multiple sequence alignment (a.k.a MSA). Based on the co-evolution hypothesis, the contact prediction for the target sequence can then be

reformulated as detecting the co-evolution nucleotide pairs in the MSA. For example, Morcos et al. [138] calculate the covariance between each pair of nucleotides, thus creating a covariance matrix as an approximation of the co-evolution. The direct coupling score (DCA score) between each pair can then be computed through different approximation methods. While Morcos et al. [138] use mean-field approximation (mfDCA), other DCA variants (e.g., [45]) use different tricks for the approximation.

DCA methods are all purely unsupervised, based on the counting frequency of the residues in MSA, and can be applied to proteins and RNA sequences. Recently, it has been shown that transformer-based protein language-models can also be unsupervised protein contact learners [156, 157], though these methods are not necessarily based on the co-evolution hypothesis.

Supervised Contact Prediction Given $\sim$ 180K known protein structures in PDB, Zhang et al. [237] train a 20M-parameters attention-based Transformer model for end-to-end prediction of protein contacts based on MSA. The attention mechanism of the model, called the Co-evolution Transformer (CoT), is specifically designed to model co-evolution by considering the outer product of representations of two positions. Such a model is only successful given a large labeled dataset of known MSA to contact mappings.

Training such a large model for RNA is unfortunately not practical as there are currently no such large datasets available. The largest of such data is at least 2-3 orders of magnitude smaller than what is available for proteins. To overcome this bottleneck, most resort to feature engineering to train smaller models. For instance, recent works [193, 243] combine DCA outputs (or similarly, covariance matrices) with other features (such as predicted secondary structures, solvent surface areas, etc.) extracted from different RNA analysis tools to train relatively small convolution networks, using only hundreds of labeled data point. Finally, Zerihun et al. [233] propose CoCoNet,

showing that the output of DCA by itself is sufficient for training such models and that oftentimes expensive additional feature extraction is not needed.

Transfer Learning We show in this chapter that the learned knowledge of a pre-trained protein contact prediction model can be effectively used for RNA contact prediction, not only removing the need for additional feature engineering and extraction but also vastly outperforming CoCoNet. Our proposed method is built upon the concept of "Transfer Learning" [39], which assumes that knowledge learned from one task is beneficial to other related tasks. Transfer learning has enabled the adaption of large pre-trained deep neural networks to new tasks with a limited number of labeled examples. This is typically done by training newly initialized layers at the end of the pre-trained network (which tends to be task-specific) using the small dataset while keeping the other layers frozen (which preserves the learned knowledge from the previous task). Only the relatively small set of parameters in the final layers will be updated, which will adapt the network to the new task.

Transfer Learning has been shown to be effective for class-level transfer in a single domain [202] and for different domains (e.g., an image classification model adapted for semantic segmentation [129]). There is also research that shows that seemingly unrelated tasks can also help each other [135]. Even models for different modalities can be transferred. For instance, Mokady et al. [136] and Lu et al. [131] show that semantic knowledge can be transferred between language and visual models.

A key challenge of RNA contact prediction is the small dataset size, which prohibits us from learning a deep model from scratch. We hypothesize (and later verify) that knowledge could be effectively transferred from a pre-trained protein contact Transformer to RNA contact prediction, enabling us to train high-performing RNA contact prediction models without the need for additional labeled or feature engineering, both of which can be prohibitively expensive. Analogies can be drawn between our

approach and research done on the cross-lingual transfer of language models [62] that adapt a pre-trained model to a new language by learning its syntax while retaining the semantic knowledge in the pre-trained model; here we are adapting a biological model pre-trained on the "language of proteins" to the "language of RNAs".

6.2.5. Methods and Setups

Protein-to-RNA Transferred Contact Prediction Model. In this section, we provide details of our model's architecture, input, and output. An overview of our approach is visualized in Figure 6.7.

MSA as Input. Our RNA contact prediction model relies on the CoT model, which takes protein MSA as the input. Thus, we need to adapt or map the RNA language, which is comprised of nucleotides, to the protein language, which is comprised of amino acids. Specifically, suppose our target RNA MSA has M aligned sequences, each with the length of L nucleotides. Then, the RNA MSA can be represented as a $M \times L$ matrix, with each element being "A", "U", "C"', "G", "-", where "-" denotes a gap in the alignment. As the CoT embedding layer recognizes only symbols corresponding to amino acids and not nucleotides, we assign each type of nucleotide in the RNA MSA to an amino acid symbol. For example, we could take a random translation from "A", "U", "C"', "G" to "H", "E", "T"', "L", to get the following translation:

"A" (Adenine) $\rightarrow$ "H" (Histdine)

"U" (Uracil) $\rightarrow$ "E" (Glutamic Acid)

"C" (Cytosine) $\rightarrow$ "T" (Threonine)

"G" (Guanine) $\rightarrow$ "L" (Leucine)

As we show in our experiments, a random translation between nucleotide and

amino acid symbols would be sufficient for adapting the protein contact prediction model, CoT, to RNA contact prediction.

The Learnable Model. The CoT model has six consecutive attention blocks and one refinement block, each outputting a $L \times L \times C$ attentive feature map, where C is a hyper-parameter corresponding to the number of features being learned by the model. In the original implementation of the protein CoT, C is set to 96, and the output of each attention block (i.e., the feature map for that block) is fed into the next block (see the "Adapted Feature Extraction" row in Figure 6.7).

For our RNA contact prediction, we further attach four layers of 2D convolution (Conv2d) modules to the intermediate feature map outputs for each of the seven attention blocks described above (see the "Feature Fusion" row in Figure 6.7). We concatenate the output of the Conv2d modules for each of the seven attention blocks into one $L \times L \times (C \times 7)$ tensor and finally pass it to a classifier module with 3 Conv2d layers for contact prediction (see the "classification" row in Figure 6.7). The output of our model has shape $L \times L \times 37$, i.e., the distance between pairs of nucleotides is divided into 37 bins. Our model is trained using standard cross-entropy loss with the bins as labels. The summed probability value of the bins for a distance less than 10Å is used as the final contact prediction.

Dataset. We use a publicly-available well-curated RNA dataset used by Zerihun et al. [233]. We use the provided data split for training, validation, and testing[3]. In total, we have 56 RNAs for training and validation and 23 RNAs for testing, all from different RNA families. We set the maximum number of homology sequences in MSA to be 200, based on the limits of our GPU memory. This constraint can be alleviated if large GPUs are available.

[3]We removed 3 RNAs (RF02540, RF01998, and RF02012) whose sequences are too long for CoT.

Baseline Methods. We compare our method to several representative MSA-based methods: (1) Unsupervised methods: mfDCA and plmDCA, using the implementations from `pydca` [232], and PSICOV [89], and PLMC [72] (2) Supervised method: CoCoNet.

Note that all these baselines are variants of DCA or are based on DCA (the current trend in studying RNA contacts). Our proposed method does not rely on DCA and approaches the problem from a different angle through the transfer learning of learned knowledge from a pre-trained protein contact prediction model.

Training Details. We use the Adam optimizer [99] and cosine anneal learning rate scheduler with an initial learning rate of $1e^{-3}$. We train on an RTX-A6000 GPU using PyTorch-1.8 and CUDA-11 and search the hyper-parameters for the total training epochs among $\{100, 300, 500\}$ and batch sizes among $\{4, 8, 12, 16\}$.

We randomly divide the 56 RNAs reserved for training into 47 RNAs for training and 9 RNAs for validation and use the given 23 RNAs in the test dataset for testing. The best-validated model during the training is used for testing.

Evaluation Metrics. Following the standard protocol of prior works [78, 182, 193, 243], we evaluate the precision on each RNA sequence of length L with top-L predictions of each method (PPV_L); i.e., for an RNA with a sequence of length L, we use our model to make L predictions (for all different (i, j) pairs). Among these L predictions, if K pairs are true contacts, then $\text{PPV}_L = \frac{K}{L}$. We also report results for $\text{PPV}_{0.5L}$ and $\text{PPV}_{0.3L}$.

6.2.6. Experiments

Unless specified otherwise, the results presented in this manuscript employ translation nucleotides to amino acids (AUCG $\rightarrow$ HETL) as detailed in Sec. 6.2.5.

Method	PPV_L	$PPV_{0.5L}$	$PPV_{0.3L}$
mfDCA	34.1	46.7	57.4
plmDCA	30.6	43.2	57.8
PSICOV	32.1	43.8	57.8
PLMC	33.5	45.9	57.4
CoCoNet§ (3×3)	61.6	67.7	69.1
CoCoNet§ (5×5)	61.8	65.2	67.8
CoCoNet§ (7×7)	62.4	66.6	69.2
CoCoNet§¶ $(3 \times 3) \times 2$	67.1	71.6	72.3
CoCoNet§¶ $(5 \times 5) \times 2$	67.5	71.9	75.0
CoCoNet§¶ $(7 \times 7) \times 2$	68.5	73.2	75.2
CoT-RNA† (Ours)	**73.5**	**80.6**	**83.0**
CoT-RNA‡ (average)	**72.1** ± 1.1	**79.2** ± 1.7	**81.9** ± 2.2

Table 6.6: Comparison of different RNA contact prediction methods based on MSA. §: Using the publicly released parameters by Zerihun et al. [233], which are trained using our training and validation sets. ¶: Using prior knowledge of Watson-Crick pairs is used. †: Our models trained using only the training set, selected based on the best validation and evaluated on the testing set. ‡: We repeat the experiments four times using different random training and validation splits and report the mean and standard deviation of the results on the test dataset.

Main Results. We first compare our method to those only using MSA as input to evaluate the contribution of the pre-trained protein Transformer to RNA contact prediction. Unsupervised algorithms like mfDCA, plmDCA, PSICOV, and PLMC are based on covariance analysis. CoCoNet uses DCA output as input and ground truth contact maps to train a supervised ConvNet classifier. We compare six CoCoNet configurations from Zerihun et al. [233].

Table 6.6 shows supervised models significantly outperform unsupervised baselines. Our transfer learning-based model outperforms the best CoCoNet configuration by an absolute of 5.0, 7.4, and 7.8 for PPVL, PPV0.5L, and $PPV_{0.3L}$, respectively.

CoCoNet is designed to be shallow, with a few parameters to learn, given the very limited number of available RNA contact prediction training data and features (from DCA). In contrast, by transfer learning of CoT, a large pre-trained protein contact prediction model, our model is much larger and deeper and can learn more diverse

features through the multi-layer attentions of CoT contain. To investigate whether the improvement of our model over CoCoNet is mainly based on its capacity or the learned knowledge of the pre-trained protein language-model, we also implement a Deep-CoCoNet, which takes the same inputs as CoCoNet but replaces the CoCoNet's original shallow ConvNet with a deep model similar to ours (with the same number of layers and each layer with the same number of channels, except for the first input layer). We find that Deep-CoCoNet performs much worse than the original shallow CoCoNet, possibly due to the hardness of fitting a large model that uses features with limited expressiveness (this may also explain why Zerihun et al. [233] use a shallow convolution network over deeper ones in their implementation).

While CoCoNet takes DCA as input which is a tensor of $1 \times L \times L$ (L being the RNA sequence length), our method, leveraging multi-layer CoT, has diverse attentive features of shape $(7 \times 96) \times L \times L$ (There are 7 layers in CoT and each layer outputs 96 channels/features). These diverse features allow us to learn deeper and larger models that can better generalize.

Common Transfer Learning Strategies. We investigate three common transfer learning strategies (see Figure 6.8 for an overview) and show that they are not well-suited for this task:

- **Using CoT directly (CoT directly).** By adapting the embedding to map RNA nucleotides to protein amino acids, the pre-trained CoT can directly output a prediction of a distance map for RNA contact prediction without any model modifications.

- **Fine-tuning the classification block of CoT (CoT cls fine-tuned).** A typical approach in transfer learning is to fine-tune the last few layers of a model. CoT has six attention blocks followed by a final ResNet block for prediction.

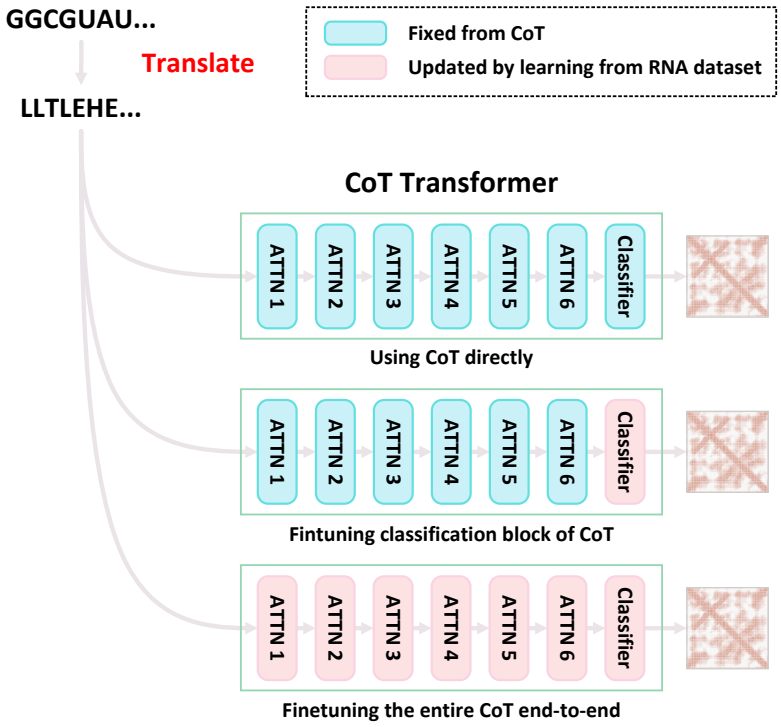

Figure 6.8: Common baselines for transferring protein CoT to RNA contact prediction.

We attach a new classification block while keeping others fixed.

- **Fine-tuning the entire CoT end-to-end (CoT end-to-end).** Another common protocol for transfer learning is to fine-tune the entire pre-trained model end-to-end. We update all parameters in the pre-trained protein CoT by the RNA training set.

Table 6.7 shows the performance of these methods on our dataset. With a PPV_L of 30.4, the direct use of CoT (CoT directly) without any learning is shown to be inefficient. This suggests that learned protein knowledge by itself cannot be successfully transferred to RNA tasks without some fine-tuning. The results for transfer learning through fine-tuning the classification block of CoT (CoT cls fine-tuned) are considerably better, being competitive with the mfDCA baseline. These results suggest that tuning the attention features in the last layer of CoT enables the

Method	PPV_L	$PPV_{0.5L}$	$PPV_{0.3L}$
mfDCA (baseline)	34.1	46.7	57.4
CoT directly	30.4	33.1	34.0
CoT cls fine-tuned	38.3	41.6	41.2
CoT end-to-end	36.2	43.2	46.6
Ours	**73.5**	**80.6**	**83.0**

Table 6.7: Common transfer learning strategies applied to CoT.

transfer of knowledge to the RNA tasks to some extent. However, as this configuration ignores the attention features from the other layers, it performs significantly worse than our method, suggesting that these features also play an important role in contact prediction and need to be tuned for RNA contact prediction. Finally, the end-to-end model, which updates all the parameters in CoT (CoT end-to-end), performs similarly to the last variant. Though this model does not ignore any part of the CoT, it requires the tuning of ∼20M parameters. With only 56 RNA training points, the model is likely to over-fit.

From these experiments, we can conclude that the effective transfer of protein CoT to the RNA contact prediction task requires (1) adapting some of the parameters of CoT to the new task, (2) leveraging the multi-layer attention features from CoT, and (3) making the number of learnable parameters "small" and proportional to the size of the training set for the new task. These findings lead to our final model design described in Section 6.2.5 that leverages multi-layer attention features from CoT and learns an appropriate number of parameters for our small training set.

Feature Fusion Design Choices. We examine various strategies for combining attention features from different CoT layers/blocks. Our model, as shown in Figure 6.7, employs multi-branch networks (each with 4 ConvNet layers) followed by a shared 3-layer ConvNet classification block. Each branch network separately processes the attention features of CoT at each layer, before being fused and passed into the

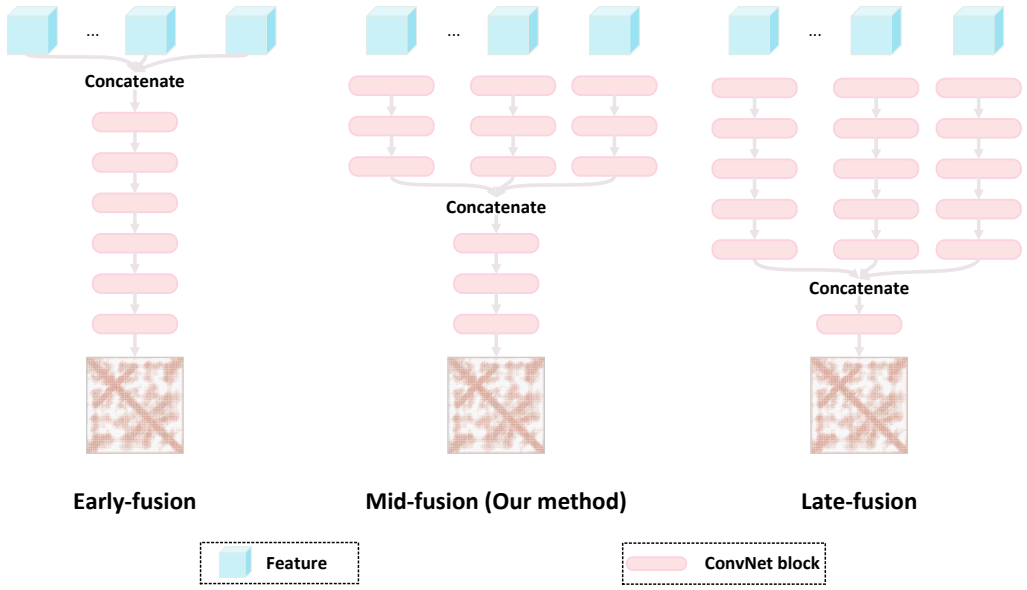

Figure 6.9: Different feature fusion strategies. Our final model uses the *mid-fusion* design.

Method	PPV_L	$PPV_{0.5L}$	$PPV_{0.3L}$
early-fusion	71.8	79.5	82.4
mid-fusion (Ours)	**73.5**	**80.6**	**83.0**
late-fusion	72.0	77.7	79.4

Table 6.8: Comparison of different feature fusion designs. We modify the number of channels in each layer so that all three models have a similar number of parameters for fair comparison.

classification block (termed *mid-fusion* design). Other designs include *early-fusion* and *late-fusion*. Early-fusion concatenates all CoT features from different layers and processes them using a single shared network. Late-fusion has separate branch networks for each CoT layer's features before being merged at the very end, followed by a single classification layer. Figure 6.9 provides a schematic diagram of these three fusion strategies.

To make the comparison of these three designs fair, we modify the number of channels in each layer so that the three models have a similar number of parameters. As shown in Table 6.8, while all design choices work well, *mid-fusion* has the best

Method	PPV_L	$PPV_{0.5L}$	$PPV_{0.3L}$
Ours (small)	71.1	76.0	79.0
Ours	73.5	80.6	83.0
Ours (large)	**78.3**	**82.3**	**86.1**

Table 6.9: Comparison of our model with different sizes. While maintaining the network structures, we vary the number of channels in each layer so that we end up with models with different numbers of parameters.

performance. It is possible that features from different layers contain different types of information that may need to be processed by different "expert" models (i.e., ConvNet branches in our model), making an early-fusion model inefficient. In both *mid-fusion* and *late-fusion*, each branch network will process the attention features of each layer separately, with *late-fusion* having a relatively smaller classification head. The overall better performance of *mid-fusion* suggests that a good design choice is to have a balanced distribution of parameters into the branch networks and the classification head.

Different Model Sizes. As discussed in Section 6.2.6, the deep-CoCoNet variant of CoCoNet under-performs compared to the shallower original CoCoNet, likely due to the limited expressiveness of input DCA features which are single channel with a shape of $[1 \times L \times L]$. In contrast, here we demonstrate that our transferred CoT model allows for learning deeper networks, with its performance improving as we increase the parameters in the transfer modules.

We create larger and smaller versions of our model by increasing and decreasing the number of channels in each layer, respectively. As shown in Table 6.9, the larger models outperform smaller ones, possibly due to the expressiveness of the CoT features from the 7 different attention blocks.

Method	PPV_L	$PPV_{0.5L}$	$PPV_{0.3L}$
AUCG → ACDE	68.6	76.9	82.5
AUCG → HETL	<u>73.5</u>	<u>80.6</u>	<u>83.0</u>
AUCG → RDSY	76.1	81.4	83.4
AUCG → KDNY	**77.3**	**84.5**	**88.6**

Table 6.10: Results of different translations/translations from nucleotides to amino acids of our transferred CoT. Bold corresponds to the best-performing translation; underline corresponds to the main translation used in the experiments.

Protein to RNA Translation Variations. We have used a random translation from RNA nucleotides to protein amino acids (e.g., "AUCG" to "HETL") in our experiments. Here, we study the effects of different translations on our model's performance.

The 20 amino acids can be categorized into four groups: (1) electrically charged, (2) polar uncharged, (3) hydrophobic, and (4) special cases. Randomly selecting one from each group generally works well (e.g., "AUCG" → "RDSY" and "AUCG" → "KDNY" in Table 6.10), indicating our framework's robustness to translation choices.

We also test possibly one of the worst translations, "AUCG" → ACDE", as it may generate unlikely amino acid chains (e.g., a string of negatively charged residues, as "D" and "E" are negatively charged) and hence CoT will have had limited exposure to such sequences during its pre-training. Though we see a relative performance drop, the results are still comparable to CoCoNet.

A learnable 4×20 nucleotide-to-amino acid embedding could yield better results but faces implementation challenges, such as requiring powerful GPUs and adapting the original CoT model's separate binary executable embedding layer to the PyTorch framework.

6.2.7. Summary

We demonstrate the effectiveness of transferring CoT, a pre-trained protein Transformer model for contact prediction, to the RNA contact prediction task using a small curated RNA dataset. Unlike hybrid methods, our approach does not use additional features extracted by RNA analysis tools (e.g., RNAcontact). Incorporating CoT features and RNA features (extracted by tools like RNAcontact) could potentially improve our method's performance.

Our findings shed light on a compelling representation transfer problem in computational structural biology; specifically, we investigate if structural patterns learned from large-scale protein datasets can be transferred to data-scarce RNA problems, particularly for structural contact predictions. Our results indicate that protein-to-RNA transfer learning can improve RNA model performance, suggesting that other pre-trained protein Transformers, such as MSA-Transformer [157] and ESM [156], could potentially be transferred to RNA for other downstream tasks.

Chapter 7

Conclusions and Discussion

The learning process for large multi-modal models typically comprises two stages: a computationally and data-intensive pre-training stage, followed by a fine-tuning stage that demands careful design for effective adaptation. Throughout this thesis, we present comprehensive solutions to facilitate the efficient and effective learning of models in both these crucial stages.

In Chapter 3, We introduce a novel optimization framework designed to enhance vision-language models through the utilization of large, pre-trained LLMs with fixed parameters. A key observation guiding our approach is that the end-to-end image-to-text pre-training process can be effectively decoupled in a backward manner. Initially, our focus lies in determining the "ideal prompt" capable of eliciting the desired target text from the LLM, a task that can be addressed through unsupervised learning. Subsequently, we align visual features with the identified prompt. To realize this decoupling, we introduce the P-Former, a model that operates analogously to a semantic sentence embedding model. The primary function of the P-Former is to predict prompts to which visual features should align. Experimental results substantiate the effectiveness of incorporating alignment loss, facilitated by the P-Former, into the framework of BLIP-2. Notably, this inclusion significantly diminishes

the performance gap observed between models trained with 4 million and 129 million image-text pairs.

In Chapter 4 introduces SimVLG, a highly efficient and streamlined pre-training framework designed for vision-language generative models. Similar to BLIP-2, SimVLG employs frozen ViT and LLM. Additionally, it incorporates a conventional Transformer architecture with token-merging capabilities, referred to as TomeFormer, serving as the crucial connector between vision and language. A notable advantage of SimVLG over BLIP-2 lies in its one-stage training approach, effectively reducing computational overhead. Even with only 1/3 to 1/10 of the computational budget required by BLIP-2, SimVLG maintains competitive performance. SimVLG serves as a testament to the possibility of achieving state-of-the-art performance in vision-language tasks without the necessity of intricate training regimens or high computational budgets. This work contributes significantly to the ongoing endeavor to develop more accessible, efficient, and potent models for comprehending and generating visual and textual information.

In Chapter 5:

- We have expanded the applicability of SimVLG to video captioning tasks by integrating the *Temporal Attentive Soft Token Merging* into its ViT. This enhancement bolsters the model's temporal modeling capabilities, resulting in a model called SimVLG-Video. Notably, this extension has proven to be effective, achieving commendable performance even in the absence of specialized video-text pre-training. Our investigation underscores the significance of a temporal module that seamlessly integrates with the well-pretrained image-text model (e.g., BLIP-2 and SimVLG), emphasizing its crucial role in contributing to this success.

- We introduce a straightforward strategy termed label hallucination, designed to streamline the efficient fine-tuning of large-capacity models using few-shot

examples from novel classes. Our approach demonstrates its effectiveness even in extreme scenarios where the labels of the base dataset and those of the novel examples are entirely disjoint. This uncomplicated procedure proves superior to prevalent strategies, such as transfer learning via fine-tuning on novel examples or linear classification atop a frozen representation. Across four well-established few-shot classification benchmarks, our method consistently outperforms current state-of-the-art approaches. Moreover, this learning paradigm of reusing pre-trained examples through pseudo-labeling is shown to be universal and applicable to VLMs, as demonstrated in BLIP [114].

- We propose a novel framework for supervised contrastive learning, complemented by an effective augmentation method utilizing prompts. This novel approach markedly boosts the performance of prompt-based language learners, outperforming recent advancements in the domain across 15 few-shot tasks. This discovery holds implications for VLMs fine-tuning, particularly as numerous VLMs employ LLMs as decoders, conditioning on visual soft-prompts. Our findings in LLMs can potentially extend to enhance results in VLM fine-tuning.

Through Chapter 3, Chapter 4, and Chapter 5, we have introduced methods for the efficient and effective training of VLMs from the pre-training stage to the fine-tuning stage. In Chapter 6, we extend our multi-modal research into bioinformatics. Drawing an analogy between image-text pairs and T-cell peptide pairs, we construct a T-cell-peptide interaction model, holding significant implications for human immunity recognition. Furthermore, we demonstrate that the acquired structural patterns in a protein Transformer can be transferred to RNA-related tasks. Given the scarcity of available RNA data in comparison to proteins, our research opens avenues for future investigations into RNA modeling.

Bibliography

[1] Arman Afrasiyabi, Jean-François Lalonde, and Christian Gagné. Associative alignment for few-shot image classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[2] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8948–8957, 2019.

[3] 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.

[4] Josephine Alba, Lorenzo Di Rienzo, Edoardo Milanetti, Oreste Acuto, and Marco D'Abramo. Molecular dynamics simulations reveal canonical conformations in different pmhc/tcr interactions. *Cells*, 9(4):942, 2020.

[5] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[6] Segun Taofeek Aroyehun and Alexander Gelbukh. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 90–97, 2018.

[7] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[8] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021.

[9] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in Neural Information Processing Systems*, 35:32897–32912, 2022.

[10] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[11] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019.

[12] Luca Bertinetto, Joao F. Henriques, Philip Torr, and Andrea Vedaldi. Meta-

learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019.

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

[14] Michal J Boniecki, Grzegorz Lach, Wayne K Dawson, Konrad Tomala, Pawel Lukasz, Tomasz Soltysinski, Kristian M Rother, and Janusz M Bujnicki. Simrna: a coarse-grained method for rna folding simulations and 3d structure prediction. *Nucleic acids research*, 44(7):e63–e63, 2016.

[15] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[16] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013.

[17] Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. Blast+: architecture and applications. *BMC bioinformatics*, 10(1):1–9, 2009.

[18] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new

model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[19] Andrea Cavalli, Xavier Salvatella, Christopher M Dobson, and Michele Vendruscolo. Protein structure determination from nmr chemical shifts. *Proceedings of the National Academy of Sciences*, 104(23):9615–9620, 2007.

[20] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200, 2011.

[21] Feilong Chen, Minglun Han, Haozhi Zhao, Qingyang Zhang, Jing Shi, Shuang Xu, and Bo Xu. X-llm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages, 2023.

[22] Min-Hung Chen, Baopu Li, Yingze Bao, Ghassan AlRegib, and Zsolt Kira. Action segmentation with joint self-supervised temporal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9454–9463, 2020.

[23] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems*, 2020.

[24] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.

[25] 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 *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX*, pages 104–120. Springer, 2020.

[26] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[27] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR, 2021.

[28] Marlid Cruz-Ramos and Jesús García-Foncillas. Car-t cell and personalized medicine. *Translational Research and Onco-Omics Applications in the Era of Cancer Personal Genomics*, pages 131–145, 2019.

[29] Wenliang Dai, Lu Hou, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. Enabling multimodal generation on clip via vision-language knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2383–2395, 2022.

[30] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023.

[31] Rhiju Das, John Karanicolas, and David Baker. Atomic accuracy in predicting and designing noncanonical rna structure. *Nature methods*, 7(4):291–294, 2010.

[32] Pradyot Dash, Andrew J Fiore-Gartland, Tomer Hertz, George C Wang, Shalini Sharma, Aisha Souquette, Jeremy Chase Crawford, E Bridie Clemens, Thi HO

Nguyen, Katherine Kedzierska, et al. Quantifiable predictive features define epitope-specific t cell receptor repertoires. *Nature*, 547(7661):89–93, 2017.

[33] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[34] Mark M Davis and Pamela J Bjorkman. T-cell antigen receptor genes and t-cell recognition. *Nature*, 334(6181):395–402, 1988.

[35] Nicolas De Neuter, Wout Bittremieux, Charlie Beirnaert, Bart Cuypers, Aida Mrzic, Pieter Moris, Arvid Suls, Viggo Van Tendeloo, Benson Ogunjimi, Kris Laukens, et al. On the feasibility of mining cd8+ t cell receptor patterns underlying immunogenic peptide recognition. *Immunogenetics*, 70(3):159–168, 2018.

[36] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[37] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186. Association for Computational Linguistics, June 2019.

[38] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *International Conference on Learning Representations*, 2020.

[39] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655. PMLR, 2014.

[40] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *Proceedings of the 31st International Conference on Machine Learning*, pages 647–655. PMLR, 22–24 Jun 2014.

[41] 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. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[42] Zi-Yi Dou, Aishwarya Kamath, Zhe Gan, Pengchuan Zhang, Jianfeng Wang, Linjie Li, Zicheng Liu, Ce Liu, Yann LeCun, Nanyun Peng, Jianfeng Gao, and Lijuan Wang. Coarse-to-fine vision-language pre-training with fusion in the backbone. In *Advances in Neural Information Processing Systems*, 2022.

[43] Zi-Yi Dou, Yichong Xu, Zhe Gan, Jianfeng Wang, Shuohang Wang, Lijuan Wang, Chenguang Zhu, Pengchuan Zhang, Lu Yuan, Nanyun Peng, et al. An empirical study of training end-to-end vision-and-language transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18166–18176, 2022.

[44] Ryan Ehrlich, Larisa Kamga, Anna Gil, Katherine Luzuriaga, Liisa K Selin, and Dario Ghersi. Swarmtcr: a computational approach to predict the specificity of t cell receptors. *BMC bioinformatics*, 22(1):1–14, 2021.

[45] Magnus Ekeberg, Cecilia Lövkvist, Yueheng Lan, Martin Weigt, and Erik Aurell. Improved contact prediction in proteins: using pseudolikelihoods to infer potts models. *Physical Review E*, 87(1):012707, 2013.

[46] Manel Esteller. Non-coding rnas in human disease. *Nature reviews genetics*, 12 (12):861–874, 2011.

[47] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. *arXiv preprint arXiv:2211.07636*, 2022.

[48] Nanyi Fei, Zhiwu Lu, Tao Xiang, and Songfang Huang. {MELR}: Meta-learning via modeling episode-level relationships for few-shot learning. In *International Conference on Learning Representations*, 2021.

[49] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1126–1135, 2017.

[50] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[51] Andrew Fire, SiQun Xu, Mary K Montgomery, Steven A Kostas, Samuel E Driver, and Craig C Mello. Potent and specific genetic interference by double-stranded rna in caenorhabditis elegans. *Nature*, 391(6669):806–811, 1998.

[52] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 1607–1616. PMLR, 2018.

[53] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. *Advances in Neural Information Processing Systems*, 33:6616–6628, 2020.

[54] Zhe Gan, Linjie Li, Chunyuan Li, Lijuan Wang, Zicheng Liu, Jianfeng Gao, et al. Vision-language pre-training: Basics, recent advances, and future trends. *Foundations and Trends® in Computer Graphics and Vision*, 14(3–4):163–352, 2022.

[55] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Association for Computational Linguistics (ACL)*, 2021.

[56] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[57] Yizhao Gao, Nanyi Fei, Guangzhen Liu, Zhiwu Lu, Tao Xiang, and Songfang Huang. Contrastive prototype learning with augmented embeddings for few-shot learning. *arXiv preprint arXiv:2101.09499*, 2021.

[58] Sofie Gielis, Pieter Moris, Wout Bittremieux, Nicolas De Neuter, Benson Ogunjimi, Kris Laukens, and Pieter Meysman. Detection of enriched t cell epitope specificity in full t cell receptor sequence repertoires. *Frontiers in immunology*, page 2820, 2019.

[59] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[60] Robert M Glaeser. How good can cryo-em become? *Nature methods*, 13(1): 28–32, 2016.

[61] Jacob Glanville, Huang Huang, Allison Nau, Olivia Hatton, Lisa E Wagar, Florian Rubelt, Xuhuai Ji, Arnold Han, Sheri M Krams, Christina Pettus, et al. Identifying specificity groups in the t cell receptor repertoire. *Nature*, 547(7661): 94–98, 2017.

[62] Evangelia Gogoulou, Ariel Ekgren, Tim Isbister, and Magnus Sahlgren. Cross-lingual transfer of monolingual models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 948–955, Marseille, France, June 2022. European Language Resources Association.

[63] Hani Goodarzi, Xuhang Liu, Hoang CB Nguyen, Steven Zhang, Lisa Fish, and Sohail F Tavazoie. Endogenous trna-derived fragments suppress breast cancer progression via ybx1 displacement. *Cell*, 161(4):790–802, 2015.

[64] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.

[65] Aurelien Grosdidier, Vincent Zoete, and Olivier Michielin. Swissdock, a protein-small molecule docking web service based on eadock dss. *Nucleic acids research*, 39(suppl_2):W270–W277, 2011.

[66] Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9101–9110, 2020.

[67] Arezoo Hatefi, Xuan-Son Vu, Monowar Bhuyan, and Frank Drewes. Cformer: Semi-supervised text clustering based on pseudo labeling. In *Proceedings of the*

30th ACM International Conference on Information & Knowledge Management, pages 3078–3082, 2021.

[68] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[69] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[70] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[71] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[72] Thomas A Hopf, John B Ingraham, Frank J Poelwijk, Charlotta PI Schärfe, Michael Springer, Chris Sander, and Debora S Marks. Mutation effects predicted from sequence co-variation. *Nature biotechnology*, 35(2):128–135, 2017.

[73] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*, 2020.

[74] Zhicheng Huang, Zhaoyang Zeng, Yupan Huang, Bei Liu, Dongmei Fu, and Jianlong Fu. Seeing out of the box: End-to-end pre-training for vision-language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12976–12985, 2021.

[75] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[76] Yiren Jian and Chongyang Gao. Metapix: Domain transfer for semantic segmentation by meta pixel weighting. *Image and Vision Computing*, 116:104334, 2021. ISSN 0262-8856. doi: https://doi.org/10.1016/j.imavis.2021.104334. URL `https://www.sciencedirect.com/science/article/pii/S0262885621002390`.

[77] Yiren Jian and Lorenzo Torresani. Label hallucination for few-shot classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7005–7014, 2022.

[78] Yiren Jian, Xiaonan Wang, Jaidi Qiu, Huiwen Wang, Zhichao Liu, Yunjie Zhao, and Chen Zeng. Direct: Rna contact predictions by integrating structural patterns. *BMC bioinformatics*, 20(1):1–12, 2019.

[79] Yiren Jian, Chongyang Gao, and Soroush Vosoughi. Non-linguistic supervision for contrastive learning of sentence embeddings. In *Advances in Neural Information Processing Systems*, 2022.

[80] Yiren Jian, Chongyang Gao, and Soroush Vosoughi. Contrastive learning for prompt-based few-shot language learners. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2022.

[81] Yiren Jian, Chongyang Gao, and Soroush Vosoughi. Embedding hallucination for few-shot language learning. In *Proceedings of the 2022 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2022.

[82] Yiren Jian, Chongyang Gao, and Soroush Vosoughi. Non-linguistic supervision for contrastive learning of sentence embeddings. In *Advances in Neural Information Processing Systems*, 2022.

[83] Yiren Jian, Erik Kruus, and Martin Renqiang Min. T-cell receptor-peptide interaction prediction with physical model augmented pseudo-labeling. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 3090–3097, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539075. URL https://doi.org/10.1145/3534678.3539075.

[84] Yiren Jian, Chongyang Gao, and Soroush Vosoughi. Bootstrapping vision-language learning with decoupled language pre-training. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=8Kch0ILfQH.

[85] Yiren Jian, Chongyang Gao, Chen Zeng, Yunjie Zhao, and Soroush Vosoughi. Knowledge from large-scale protein contact prediction models can be transferred to the data-scarce rna contact prediction task, 2023. URL https://arxiv.org/abs/2302.06120.

[86] Yiren Jian, Tingkai Liu, Yunzhe Tao, Soroush Vosoughi, and Hongxia Yang. Simvlg: Simple and efficient pretraining of visual language generative models. *arXiv preprint arXiv:2310.03291*, 2023.

[87] Woojeong Jin, Yu Cheng, Yelong Shen, Weizhu Chen, and Xiang Ren. A good prompt is worth millions of parameters: Low-resource prompt-based learning

for vision-language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2763–2775, 2022.

[88] Emmi Jokinen, Jani Huuhtanen, Satu Mustjoki, Markus Heinonen, and Harri Lähdesmäki. Predicting recognition between t cell receptors and epitopes with tcrgp. *PLoS computational biology*, 17(3):e1008814, 2021.

[89] David T Jones, Daniel WA Buchan, Domenico Cozzetto, and Massimiliano Pontil. Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190, 2012.

[90] Magdalena A Jonikas, Randall J Radmer, Alain Laederach, Rhiju Das, Samuel Pearlman, Daniel Herschlag, and Russ B Altman. Coarse-grained modeling of large rna molecules with knowledge-based potentials and structural filters. *Rna*, 15(2):189–199, 2009.

[91] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[92] Vanessa Isabell Jurtz, Leon Eyrich Jessen, Amalie Kai Bentzen, Martin Closter Jespersen, Swapnil Mahajan, Randi Vita, Kamilla Kjærgaard Jensen, Paolo Marcatili, Sine Reker Hadrup, Bjoern Peters, et al. Nettcr: sequence-based prediction of tcr binding to peptide-mhc complexes using convolutional neural networks. *BioRxiv*, page 433706, 2018.

[93] Morten Källberg, Haipeng Wang, Sheng Wang, Jian Peng, Zhiyong Wang, Hui

Lu, and Jinbo Xu. Template-based protein structure modeling using the raptorx web server. *Nature protocols*, 7(8):1511–1522, 2012.

[94] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021.

[95] Akbar Karimi, Leonardo Rossi, and Andrea Prati. AEDA: an easier data augmentation technique for text classification. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 2748–2754. Association for Computational Linguistics, 2021. URL `https://aclanthology.org/2021.findings-emnlp.234`.

[96] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[97] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf`.

[98] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language trans-

former without convolution or region supervision. In *International Conference on Machine Learning*, pages 5583–5594. PMLR, 2021.

[99] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

[100] Gregory R. Koch. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, 2015.

[101] 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*, 123:32–73, 2017.

[102] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[103] Michelle Krogsgaard and Mark M Davis. How t cells' see'antigen. *Nature immunology*, 6(3):239–245, 2005.

[104] Andrey Krokhotin, Kevin Houlihan, and Nikolay V Dokholyan. ifoldrna v2: folding rna with constraints. *Bioinformatics*, 31(17):2891–2893, 2015.

[105] Nicole L La Gruta, Stephanie Gras, Stephen R Daley, Paul G Thomas, and Jamie Rossjohn. Understanding the drivers of mhc restriction of t cell receptors. *Nature Reviews Immunology*, 18(7):467–478, 2018.

[106] Michalis Lazarou, Yannis Avrithis, and Tania Stathaki. Iterative label cleaning for transductive and semi-supervised few-shot learning. *arXiv preprint arXiv:2012.07962*, 2020.

[107] Andrew Leaver-Fay, Michael Tyka, Steven M Lewis, Oliver F Lange, James Thompson, Ron Jacak, Kristian W Kaufman, P Douglas Renfrew, Colin A Smith, Will Sheffler, et al. Rosetta3: an object-oriented software suite for the simulation and design of macromolecules. In *Methods in enzymology*, volume 487, pages 545–574. Elsevier, 2011.

[108] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

[109] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[110] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021.

[111] Chenliang Li, Haiyang Xu, Junfeng Tian, Wei Wang, Ming Yan, Bin Bi, Jiabo Ye, He Chen, Guohai Xu, Zheng Cao, Ji Zhang, Songfang Huang, Fei Huang, Jingren Zhou, and Luo Si. mPLUG: Effective and efficient vision-language learning by cross-modal skip-connections. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7241–7259. Association for Computational Linguistics, December 2022.

[112] Dongxu Li, Junnan Li, Hung Le, Guangsen Wang, Silvio Savarese, and Steven C. H. Hoi. Lavis: A library for language-vision intelligence, 2022.

[113] 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*, 34:9694–9705, 2021.

[114] 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.

[115] 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.

[116] Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023.

[117] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[118] Liunian Harold Li, Haoxuan You, Zhecan Wang, Alireza Zareian, Shih-Fu Chang, and Kai-Wei Chang. Unsupervised vision-and-language pre-training without parallel images and captions. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5339–5350, 2021.

[119] Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. Unimo: Towards unified-modal understanding and generation via cross-modal contrastive learning. In *Proceedings of the 59th*

Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 2592–2607, 2021.

[120] Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. In *Advances in Neural Information Processing Systems*, 2019.

[121] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. Springer, 2020.

[122] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[123] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic level protein structure with a language model. *bioRxiv*, 2022.

[124] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, 2020.

[125] Haogeng Liu, Qihang Fan, Tingkai Liu, Linjie Yang, Yunzhe Tao, Huaibo Huang, Ran He, and Hongxia Yang. Video-teller: Enhancing cross-modal generation with fusion and decoupling. *arXiv preprint arXiv:2310.04991*, 2023.

[126] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*, 2023.

[127] Shikun Liu, Linxi Fan, Edward Johns, Zhiding Yu, Chaowei Xiao, and Anima Anandkumar. Prismer: A vision-language model with an ensemble of experts. *arXiv preprint arXiv:2303.02506*, 2023.

[128] Tingkai Liu, Yunzhe Tao, Haogeng Liu, Qihang Fan, Ding Zhou, Huaibo Huang, Ran He, and Hongxia Yang. Video-csr: Complex video digest creation for visual-language models. *arXiv preprint arXiv:2310.05060*, 2023.

[129] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[130] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.

[131] Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Pretrained transformers as universal computation engines. *arXiv preprint arXiv:2103.05247*, 2021.

[132] Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Minghui Qiu, Pengcheng Lu, Tao Wang, and Zhongyu Wei. Valley: Video assistant with large language model enhanced ability. *arXiv preprint arXiv:2306.07207*, 2023.

[133] Haiyun Ma, Xinyu Jia, Kaiming Zhang, and Zhaoming Su. Cryo-em advances in rna structure determination. *Signal Transduction and Targeted Therapy*, 7 (1):1–6, 2022.

[134] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[135] Elliot Meyerson and Risto Miikkulainen. The traveling observer model: Multi-task learning through spatial variable embeddings. In *ICLR*, 2021.

[136] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021.

[137] Alessandro Montemurro, Viktoria Schuster, Helle Rus Povlsen, Amalie Kai Bentzen, Vanessa Jurtz, William D Chronister, Austin Crinklaw, Sine R Hadrup, Ole Winther, Bjoern Peters, et al. Nettcr-2.0 enables accurate prediction of tcr-peptide binding by using paired tcrα and β sequence data. *Communications biology*, 4(1):1–13, 2021.

[138] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011.

[139] Salman Khan Muhammad Maaz, Hanoona Rasheed and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *ArXiv 2306.05424*, 2023.

[140] Anna Paola Muntoni, Andrea Pagnani, Martin Weigt, and Francesco Zamponi. adabmdca: adaptive boltzmann machine learning for biological sequences. *BMC bioinformatics*, 22(1):1–19, 2021.

[141] Eric P Nawrocki and Sean R Eddy. Infernal 1.1: 100-fold faster rna homology searches. *Bioinformatics*, 29(22):2933–2935, 2013.

[142] Vicente Ordonez, Girish Kulkarni, and Tamara Berg. Im2text: Describing images using 1 million captioned photographs. *Advances in neural information processing systems*, 24, 2011.

[143] Boris N. Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018.

[144] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3764–3773, 2020.

[145] Tatu Pantsar and Antti Poso. Binding affinity via docking: fact and fiction. *Molecules*, 23(8):1899, 2018.

[146] Isabel Papadimitriou and Dan Jurafsky. Learning music helps you read: Using transfer to study linguistic structure in language models. In *EMNLP*, pages 6829–6839, 01 2020. doi: 10.18653/v1/2020.emnlp-main.554.

[147] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[148] William R Pearson and David J Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448, 1988.

[149] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11557–11568, June 2021.

[150] Hieu Pham, Zihang Dai, Qizhe Xie, Minh-Thang Luong, and Quoc V. Le. Meta pseudo labels. *arXiv preprint arXiv:2003.10580*, 2021.

[151] Brian G Pierce, Kevin Wiehe, Howook Hwang, Bong-Hyun Kim, Thom Vreven, and Zhiping Weng. Zdock server: interactive docking prediction of protein–protein complexes and symmetric multimers. *Bioinformatics*, 30(12):1771–1773, 2014.

[152] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[153] Mariusz Popenda, Marta Szachniuk, Maciej Antczak, Katarzyna J Purzycka, Piotr Lukasiak, Natalia Bartol, Jacek Blazewicz, and Ryszard W Adamiak. Automated 3d structure composition for large rnas. *Nucleic acids research*, 40 (14):e112–e112, 2012.

[154] 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.

[155] Jathushan Rajasegaran, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Mubarak Shah. Self-supervised knowledge distillation for few-shot learning. *arXiv preprint arXiv:2006.09785*, 2020.

[156] Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. In *International Conference on Learning Representations*, 2021.

[157] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR, 2021.

[158] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[159] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[160] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature methods*, 9(2):173–175, 2012.

[161] Mengye Ren, Sachin Ravi, Eleni Triantafillou, Jake Snell, Kevin Swersky, Josh B. Tenenbaum, Hugo Larochelle, and Richard S. Zemel. Meta-learning for semi-supervised few-shot classification. In *International Conference on Learning Representations*, 2018.

[162] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[163] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024, 2017.

[164] Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*, 2020.

[165] Mamshad Nayeem Rizve, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Exploring complementary strengths of invariant and equivariant representations for few-shot learning. *arXiv preprint arXiv:2103.01315*, 2021.

[166] Jamie Rossjohn, Stephanie Gras, John J Miles, Stephen J Turner, Dale I Godfrey, and James McCluskey. T cell antigen receptor recognition of antigen-presenting molecules. *Annual review of immunology*, 33:169–200, 2015.

[167] Magdalena Rother, Kaja Milanowska, Tomasz Puton, Jaroslaw Jeleniewicz, Kristian Rother, and Janusz M Bujnicki. Moderna server: an online tool for modeling rna 3d structures. *Bioinformatics*, 27(17):2441–2442, 2011.

[168] Karina B Santos, Isabella A Guedes, Ana LM Karl, and Laurent E Dardenne. Highly flexible ligand docking: benchmarking of the dockthor program on the leads-pep protein–peptide data set. *Journal of Chemical Information and Modeling*, 60(2):667–683, 2020.

[169] Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *EACL*, 2021.

[170] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[171] Stefan E Seemann, Peter Menzel, Rolf Backofen, and Jan Gorodkin. The petfold and petcofold web servers for intra-and intermolecular structures of multiple rna sequences. *Nucleic acids research*, 39(suppl_2):W107–W111, 2011.

[172] Andrew Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577: 1–5, 01 2020. doi: 10.1038/s41586-019-1923-7.

[173] 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)*, pages 2556–2565, 2018.

[174] Upasna Sharma, Colin C Conine, Jeremy M Shea, Ana Boskovic, Alan G Derr, Xin Y Bing, Clemence Belleannee, Alper Kucukural, Ryan W Serra, Fengyun Sun, et al. Biogenesis and function of trna fragments during sperm maturation and fertilization in mammals. *Science*, 351(6271):391–396, 2016.

[175] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can CLIP benefit vision-

and-language tasks? In *International Conference on Learning Representations*, 2022.

[176] Zhiqiang Shen, Zechun Liu, Jie Qin, Marios Savvides, and Kwang-Ting Cheng. Partial is better than all: Revisiting fine-tuning strategy for few-shot learning. *CoRR*, abs/2102.03983, 2021.

[177] Mang Shi, Xian-Dan Lin, Jun-Hua Tian, Liang-Jun Chen, Xiao Chen, Ci-Xiu Li, Xin-Cheng Qin, Jun Li, Jian-Ping Cao, John-Sebastian Eden, et al. Redefining the invertebrate rna virosphere. *Nature*, 540(7634):539–543, 2016.

[178] Ya-Zhou Shi, Lei Jin, Feng-Hua Wang, Xiao-Long Zhu, and Zhi-Jie Tan. Predicting 3d structure, flexibility, and stability of rna hairpins in monovalent and divalent ion solutions. *Biophysical journal*, 109(12):2654–2665, 2015.

[179] Mikhail Shugay, Dmitriy V Bagaev, Ivan V Zvyagin, Renske M Vroomans, Jeremy Chase Crawford, Garry Dolton, Ekaterina A Komech, Anastasiya L Sycheva, Anna E Koneva, Evgeniy S Egorov, et al. Vdjdb: a curated database of t-cell receptor sequences with known antigen specificity. *Nucleic acids research*, 46(D1):D419–D427, 2018.

[180] Christian Simon, Piotr Koniusz, Richard Nock, and Mehrtash Harandi. Adaptive subspaces for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[181] Alok Singh, Thoudam Doren Singh, and Sivaji Bandyopadhyay. Nits-vc system for vatex video captioning challenge 2020. *arXiv preprint arXiv:2006.04058*, 2020.

[182] Jaswinder Singh, Kuldip Paliwal, Thomas Litfin, Jaspreet Singh, and Yaoqi Zhou. Predicting rna distance-based contact maps by integrated deep learning

on physics-inferred secondary structure and evolutionary-derived mutational coupling. *Bioinformatics*, 38(16):3900–3910, 2022.

[183] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

[184] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, pages 596–608, 2020.

[185] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33, 2020.

[186] Ido Springer, Hanan Besser, Nili Tickotsky-Moskovitz, Shirit Dvorkin, and Yoram Louzoun. Prediction of specific tcr-peptide binding from large dictionaries of tcr-peptide pairs. *Frontiers in immunology*, 11:1803, 2020.

[187] Ido Springer, Nili Tickotsky, and Yoram Louzoun. Contribution of t cell receptor alpha and beta cdr3, mhc typing, v and j genes to peptide binding prediction. *Frontiers in immunology*, 12, 2021.

[188] Gerald Stubbs, Stephen Warren, and Kenneth Holmes. Structure of rna and rna binding site in tobacco mosaic virus from 4-å map calculated from x-ray fibre diagrams. *Nature*, 267(5608):216–221, 1977.

[189] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020.

[190] Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. Pandagpt: One model to instruction-follow them all. *arXiv preprint arXiv:2305.16355*, 2023.

[191] Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[192] Saisai Sun, Qi Wu, Zhenling Peng, and Jianyi Yang. Enhanced prediction of rna solvent accessibility with long short-term memory neural networks and improved sequence profiles. *Bioinformatics*, 35(10):1686–1691, 2019.

[193] Saisai Sun, Wenkai Wang, Zhenling Peng, and Jianyi Yang. Rna inter-nucleotide 3d closeness prediction by deep residual neural networks. *Bioinformatics*, 37(8): 1093–1098, 2021.

[194] Derek Tam, Rakesh R Menon, Mohit Bansal, Shashank Srivastava, and Colin Raffel. Improving and simplifying pattern exploiting training. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

[195] Hao Tan and Mohit Bansal. 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, 2019.

[196] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B. Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: A good embedding is all you need? In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, 2020.

[197] Nili Tickotsky, Tal Sagiv, Jaime Prilusky, Eric Shifrut, and Nir Friedman. Mcpas-tcr: a manually curated catalogue of pathology-associated t cell receptor sequences. *Bioinformatics*, 33(18):2924–2929, 2017.

[198] Yao Tong, Jiayin Wang, Tian Zheng, Xuanping Zhang, Xiao Xiao, Xiaoyan Zhu, Xin Lai, and Xiang Liu. Sete: Sequence-based ensemble learning approach for tcr epitope binding prediction. *Computational Biology and Chemistry*, 87: 107281, 2020.

[199] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

[200] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[201] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

[202] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[203] Jiaan Wang, Zhixu Li, Qiang Yang, Jianfeng Qu, Zhigang Chen, Qingsheng Liu, and Guoping Hu. Sportssum2. 0: Generating high-quality sports news from live text commentary. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3463–3467, 2021.

[204] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. GIT: A generative image-to-text transformer for vision and language. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

[205] Jun Wang, Jian Wang, Yanzhao Huang, and Yi Xiao. 3drna v2. 0: An updated web server for rna 3d structure prediction. *International Journal of Molecular Sciences*, 20(17):4116, 2019.

[206] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022.

[207] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.

[208] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4581–4591, 2019.

[209] Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu. Instance credibility inference for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[210] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan

Cao. SimVLM: Simple visual language model pretraining with weak supervision. In *International Conference on Learning Representations*, 2022.

[211] Benjamin Webb and Andrej Sali. Comparative protein structure modeling using modeller. *Current protocols in bioinformatics*, 54(1):5–6, 2016.

[212] Anna Weber, Jannis Born, and María Rodriguez Martínez. Titan: T-cell receptor specificity prediction with bimodal attention networks. *Bioinformatics*, 37(Supplement_1):i237–i244, 2021.

[213] Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. In *International Conference on Learning Representations*, 2021.

[214] Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6383–6389, Hong Kong, China, November 2019. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/D19-1670`.

[215] Martin Weigt, Robert A White, Hendrik Szurmant, James A Hoch, and Terence Hwa. Identification of direct residue contacts in protein–protein interaction by message passing. *Proceedings of the National Academy of Sciences*, 106(1):67–72, 2009.

[216] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[217] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[218] Xiao Xu, Chenfei Wu, Shachar Rosenman, Vasudev Lal, Wanxiang Che, and Nan Duan. Bridgetower: Building bridges between encoders in vision-language representation learning. *arXiv preprint arXiv:2206.08657*, 2022.

[219] Xiaojun Xu, Peinan Zhao, and Shi-Jie Chen. Vfold: a web server for rna structure and folding thermodynamics prediction. *PloS one*, 9(9):e107504, 2014.

[220] Hongwei Xue, Yupan Huang, Bei Liu, Houwen Peng, Jianlong Fu, Houqiang Li, and Jiebo Luo. Probing inter-modality: Visual parsing with self-attention for vision-and-language pre-training. *Advances in Neural Information Processing Systems*, 34:4514–4528, 2021.

[221] Shen Yan, Tao Zhu, Zirui Wang, Yuan Cao, Mi Zhang, Soham Ghosh, Yonghui Wu, and Jiahui Yu. Videococa: Video-text modeling with zero-shot transfer from contrastive captioners, 2022.

[222] Yumeng Yan, Huanyu Tao, Jiahua He, and Sheng-You Huang. The hdock server for integrated protein–protein docking. *Nature protocols*, 15(5):1829–1852, 2020.

[223] Jianyi Yang, Renxiang Yan, Ambrish Roy, Dong Xu, Jonathan Poisson, and Yang Zhang. The i-tasser suite: protein structure and function prediction. *Nature methods*, 12(1):7–8, 2015.

[224] Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. Improved protein structure prediction using predicted

interresidue orientations. *Proceedings of the National Academy of Sciences*, 117 (3):1496–1503, 2020.

[225] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*, pages 521–539. Springer, 2022.

[226] Han-Jia Ye, Hexiang Hu, De-Chuan Zhan, and Fei Sha. Few-shot learning via embedding adaptation with set-to-set functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[227] Sung Whan Yoon, Jun Seo, and Jaekyun Moon. TapNet: Neural network augmented with task-adaptive projection for few-shot learning. In *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 7115–7123. PMLR, 09–15 Jun 2019.

[228] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

[229] Zhongjie Yu, Lin Chen, Zhongwei Cheng, and Jiebo Luo. Transmatch: A transfer-learning scheme for semi-supervised few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[230] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classi-

fiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[231] Yan Zeng, Xinsong Zhang, and Hang Li. Multi-grained vision language pre-training: Aligning texts with visual concepts. In *International Conference on Machine Learning*, pages 25994–26009. PMLR, 2022.

[232] Mehari B Zerihun, Fabrizio Pucci, Emanuel K Peter, and Alexander Schug. pydca v1. 0: a comprehensive software for direct coupling analysis of rna and protein sequences. *Bioinformatics*, 36(7):2264–2265, 2020.

[233] Mehari B Zerihun, Fabrizio Pucci, and Alexander Schug. Coconet—boosting rna contact prediction by convolutional neural networks. *Nucleic acids research*, 49(22):12661–12672, 2021.

[234] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34, 2021.

[235] Chi Zhang, Yujun Cai, Guosheng Lin, and Chunhua Shen. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[236] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

[237] He Zhang, Fusong Ju, Jianwei Zhu, Liang He, Bin Shao, Nanning Zheng, and

Tie-Yan Liu. Co-evolution transformer for protein contact prediction. *Advances in Neural Information Processing Systems*, 34:14252–14263, 2021.

[238] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[239] Manli Zhang, Jianhong Zhang, Zhiwu Lu, Tao Xiang, Mingyu Ding, and Songfang Huang. {IEPT}: Instance-level and episode-level pretext tasks for few-shot learning. In *International Conference on Learning Representations*, 2021.

[240] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12414–12424, 2021.

[241] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588, 2021.

[242] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[243] Tongchuan Zhang, Jaswinder Singh, Thomas Litfin, Jian Zhan, Kuldip Paliwal, and Yaoqi Zhou. Rnacmap: a fully automatic pipeline for predicting contact maps of rnas by evolutionary coupling analysis. *Bioinformatics*, 37(20):3494–3500, 2021.

[244] Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13278–13288, 2020.

[245] Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate Before Use: Improving Few-shot Performance of Language Models. In *International Conference on Machine Learning (ICML)*, 2021.

[246] Yunjie Zhao, Yangyu Huang, Zhou Gong, Yanjie Wang, Jianfen Man, and Yi Xiao. Automated and fast building of three-dimensional rna structures. *Scientific reports*, 2(1):1–6, 2012.

[247] Mingyang Zhou, Licheng Yu, Amanpreet Singh, Mengjiao Wang, Zhou Yu, and Ning Zhang. Unsupervised vision-and-language pre-training via retrieval-based multi-granular alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16485–16494, 2022.