UNIVERSITY OF
CAMBRIDGE
Department of Computer
Science and Technology

Integrating Cognitively-Inspired Selective Attention Cues in Small Vision-Language Models

Bianca-Mihaela Ganescu

Clare Hall

June 2025

Submitted in partial fulfillment of the requirements for the
Master of Philosophy in Advanced Computer Science

Total page count: 68

Main chapters (excluding front-matter, references and appendix): 46 pages (pp 8–53)

Main chapters word count: 14,922

Methodology used to generate that word count:

Overleaf word counter (which uses *texcount*), with the following settings in the preamble of the *.tex* file:

```
%TC:macro \cite [option:text,text]
%TC:macro \citep [option:text,text]
%TC:macro \citeauthor [option:text,text]
%TC:macro \citealp [option:text,text]
%TC:envir listing 0 1
%TC:envir minted 0 0
%TC:envir algorithm 0 1
%TC:envir algorithmic 0 0
%TC:envir table 0 1
%TC:envir tabular 0 1
%TC:macro \caption [option:text,text]
%TC:macro \footnote [option:text,text]
%TC:macro \paragraph [option:text,text]
%TC:group figure 0 1
```

Declaration

I, Bianca-Mihaela Ganescu of Clare Hall, being a candidate for the Master of Philosophy in Advanced Computer Science, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose. In preparation of this report, I adhered to the Department of Computer Science and Technology AI Policy. I am content for my report to be made available to the students and staff of the University.

Signed: 

Date: 10/06/2025

Abstract

Recent advancements in language modelling have achieved impressive results by scaling up parameters and pretraining on ever larger datasets, yet children acquire their first language from just tens of millions of words. This contrast raises fundamental questions of whether language models can learn more like humans do: efficiently, incrementally and grounded in perceptual experience. This research addresses this gap within the framework of the BabyLM Challenge Vision track, by developing a cognitively-inspired multimodal framework that learns *when* and *how* to leverage visual cues during language processing without explicit supervision.

In this work, I propose a decoder-based vision-language model with three key innovations. First, I implement a token-wise dynamic gating mechanism that learns to selectively weigh visual versus linguistic cues based on context. Second, I investigate feature modulation and channel attention techniques as solutions to limited visual information under the BabyLM Challenge Vision track constraints. Third, I explore contrastive learning auxiliary objectives for visual grounding under low-resource data regimes.

Experiments on five BabyLM Challenge benchmarks reveal task-specific benefits of my proposed framework. My base model achieves competitive or superior performance compared to multimodal baselines, despite using only global image embeddings and significantly fewer training epochs. Most significantly, statistical analysis demonstrates that the dynamic gating mechanism of my framework discovers cognitively-meaningful patterns without supervision: models learn to assign more weight to visual signals for content words (nouns, verbs, adjectives), while relying more on linguistic cues for function words (conjunctions, auxiliary verbs, particles).

However, my results also reveal that multiple constraints of the BabyLM Challenge Vision track may be unsuitable for achieving performant vision BabyLMs. Using only global image embeddings represents an information bottleneck that feature enhancement techniques are unable to resolve. Contrastive learning auxiliary objectives negatively impact performance under current constraints, whereas training models on both ungrounded and visually-grounded text data introduces training complexity and instability. Furthermore, mismatches between training data and evaluation benchmarks limit the evaluation of the models.

These findings suggest that while architectural innovations such as dynamic gating can lead to cognitively-meaningful patterns, significant constraints in visual representation, data curriculum and evaluation benchmarks must be addressed to bridge the gap between vision-language models and human language learning.

Acknowledgements

I would like to thank my supervisors, Dr Andrew Caines, Suchir Salhan and Professor Paula Buttery, as well as Diana Galvan Sosa, for their guidance, support and valuable feedback throughout this project.

I am also deeply grateful to Burak and to my parents, who provided immense support and encouragement throughout my studies at the University of Cambridge.

Contents

Chapter 1

Introduction

Large language models have achieved impressive capabilities, yet their learning process significantly contrasts with human language learning. Children learn their first language from just tens of millions of words (Warstadt et al., 2023; Gilkerson et al., 2017) with minimal supervision, whereas state-of-the-art language models require three to four magnitudes more data (Warstadt et al., 2023). This discrepancy raises fundamental questions about the nature of first language acquisition and whether language models can learn more like humans do: efficiently, incrementally and grounded in perceptual experience.

The BabyLM Challenge (Warstadt et al., 2023; Choshen et al., 2024; Charpentier et al., 2025) addresses this gap by constraining language models to train on cognitively-plausible amounts of data, approximately 100 million words reflecting the quantity a child is exposed to by adolescence (Warstadt et al., 2023; Gilkerson et al., 2017). While the text-only track has received significant attention, human language learning is inherently a multimodal process. In particular, visual experiences play a crucial role in the acquisition of early language and its expansion in the first years of life (Rose et al., 2009). This cognitive reality motivates my research in the BabyLM Challenge Vision track, where I develop a framework inspired by human selective attention that learns *when* and *how* to leverage visual cues during language processing without explicit supervision.

Specifically, I develop a cognitively-inspired multimodal framework that learns language from limited amounts of ungrounded and visually-grounded text in the context of the BabyLM Challenge Vision track. The base of my approach is a decoder-based vision-language model for which I introduce three key innovations. First, I implement a dynamic gating mechanism that learns to selectively weigh visual versus linguistic cues for each token based on context. Second, I explore several feature enhancement techniques in order to maximise the utility of limited visual information, which is a constraint of the BabyLM Challenge Vision track. Third, I investigate the impact of contrastive learning auxiliary objective functions that operate at both the sentence and word levels under low-resource constraints.

In this research, I aim to answer several key questions:

1. Can dynamic gating mechanisms be repurposed to learn vision-language fusion patterns that are cognitively-meaningful without explicit supervision? (Section 3.4)
2. Is the setup of the BabyLM Challenge Vision track optimal for multimodal learning? In particular, can architectural mechanisms compensate for the limited visual information provided by global image embeddings? (Section 3.5)
3. Do contrastive learning auxiliary objectives help or hinder small vision-language models under significant data constraints? (Section 3.6)
4. Which training and data curriculum strategies best support vision-language models in low-resource regimes? (Section 3.7)
5. Which linguistic phenomena do my models prioritise visual information for, and does this align with human word grounding? (Subsection 4.5.2 and Section 5.6)

The experiments I conduct in this work yield several important findings. Performance analysis on five BabyLM Challenge benchmarks reveals task-specific benefits of my proposed framework, with my base model achieving competitive or superior performance compared to the multimodal baselines of the Challenge, despite using only global image embeddings and significantly fewer training epochs (Section 5.1 and Subsection 5.2.1). More significantly, statistical analysis of the dynamic gating outputs shows that this mechanism is able to discover cognitively-meaningful patterns without explicit supervision: the model learns to assign more weight to visual signals for content words (nouns, verbs, adjectives) and rely more on linguistic cues for function words (conjunctions, auxiliary verbs, particles) (Section 5.6). These results suggest that vision BabyLMs do not need to be explicitly taught when to use visual information but rather that the right architecture enables them to discover principles that align with human cognition independently.

However, my results also reveal that multiple constraints of the BabyLM Challenge may be unsuitable to achieve performant vision BabyLMs. Specifically, using only global image embeddings represents an information bottleneck (Section 6.1.2). Despite exploring feature modulation and channel attention mechanisms, the model is unable to extract fine-grained visual information from a single *CLS* token (Subsection 5.2.2). The global image embeddings also seem to be insufficient for contrastive learning auxiliary objectives, which negatively impact performance under the current constraints of the Challenge (Section 5.3). Moreover, the provided training data seems to be misaligned with multiple evaluation benchmarks (Section 5.5 and Section 6.2). Additionally, the split of the training dataset between text-only and image-caption data introduces training complexity and instability (Section 5.4).

Overall, this work contributes to the broader goal of developing language models that learn more like humans, not just in terms of data, but also in their underlying mechanisms. While the results of dynamic gating show that architectural design can lead to cognitively-meaningful patterns, my findings also reveal which constraints (visual representation, data curriculum, training datasets and evaluation benchmarks) must be addressed next in order to bridge the gap between vision-language models and human language learning.

Chapter 2

Background

2.1 BabyLM Challenge

2.1.1 BabyLM Challenge Overview

The BabyLM Challenge (Warstadt et al., 2023; Choshen et al., 2024; Charpentier et al., 2025) is a shared task focused on developing computational models of first language acquisition that are *both* cognitively plausible *and* data efficient. While state-of-the-art language models are trained on vast datasets that far exceed the amount of data humans are exposed to, the BabyLM Challenge restricts training data to realistic volumes that approximate the quantity from which a child learns its first language.

The challenge proposes three different tracks that explore language learning under specific constraints: text-only, multimodal vision-and-language (referred to as the Vision track), and interaction-based. In my work, I investigate the Vision track, which incorporates image and text input to more closely simulate the multimodal nature of first language acquisition.

2.1.2 Vision Track Rules

The Vision track's rules permit implementing any model architecture, training regime and objective function that allows inference on image and text input, as long as the training dataset consists of at most 100 million words, representing an upper limit to the number of words that children are exposed to by the beginning of adolescence (Warstadt et al., 2023; Gilkerson et al., 2017). For the 2025 BabyLM Challenge, the authors set a limit of 10 training epochs in order to reduce the emphasis on access to computing resources (Charpentier et al., 2025), which I also respect in my implementation.

2.1.3 Pretraining Dataset

The BabyLM Challenge organisers provide an image-text pretraining dataset for the Vision track, which I use in my work. This dataset consists of two parts: text-only data and text-image data, each

containing approximately 50 million words.

The text-only dataset is a subset of the training data proposed for the BabyLM Challenge text-only track. The organisers argue that this dataset is cognitively plausible, consisting of child-directed speech (CHILDES (MacWhinney, 2000)), dialogue (British National Corpus (BNC) conversation section[1], Switchboard Dialog Act Corpus (Stolcke et al., 2000)), children's stories (Project Gutenberg (Gerlach and Font-Clos, 2020)), movie subtitles (OpenSubtitles (Lison and Tiedemann, 2016)) and Wikipedia[2] content.

The multimodal dataset consists of image-caption pairs selected from the Conceptual Captions 3M dataset (Sharma et al., 2018), and the MS-COCO (Lin et al., 2014) and Open Images (Kuznetsova et al., 2020) subsets of the Localized Narratives dataset (Pont-Tuset et al., 2020). The Conceptual Captions dataset consists of millions of images paired with natural language descriptions automatically scraped, cleaned and filtered from web image alt-text, while the Localized Narratives dataset contains image-caption pairs manually annotated with synchronised mouse traces that spatially ground each word or phrase to specific regions in the image. The images are provided in both raw format and as visual embeddings computed by a visual model using DINOv2 (Choshen et al., 2024; Oquab et al., 2023), a state-of-the-art unsupervised learning algorithm. I use these visual embeddings in both my training and evaluation due to computational constraints.

A breakdown of the number of words and images drawn from each data source is provided in Appendix E.

2.1.4 Vision-Language Model Baselines

The baselines used in the Vision track are the Flamingo (Alayrac et al., 2022) and GIT (Wang et al., 2022) vision-language models. Vision-language models combine an image encoder and (optionally) a text encoder with a multimodal fusion module to learn joint representations for tasks such as captioning, retrieval, and visual question answering.

The Generative Image-to-text Transformer (GIT) (Wang et al., 2022) architecture consists of an image encoder and a text decoder. The image encoder is first pretrained using a contrastive learning objective. The visual features outputted by the image encoder are linearly projected and concatenated with embedded text tokens to form the input to the decoder. The entire model is then trained using next token prediction as the objective function, where each token is predicted based on both the preceding text tokens and the visual features.

Flamingo (Alayrac et al., 2022) is a decoder-based multimodal model that interleaves text decoder layers with gated cross-attention dense blocks that incorporate visual input. First, an image encoder extracts visual features from the input images, then a Perceiver Resampler module (Jaegle et al., 2021) compresses them into a fixed number of tokens per image. These visual features are used as keys and queries in the gated cross-attention dense layers inserted between language model blocks,

[1] http://www.natcorp.ox.ac.uk
[2] https://www.wikipedia.org

where a tahn-gated learnable scalar scales each cross-attention and feed-forward sublayer to control how much visual information is fused. The proposed objective function for training the model is next token prediction.

At the time of writing, the Flamingo and GIT models provided in the BabyLM Challenge 2024 are the only publicly available baselines[3], which were trained for 20 epochs on the Vision track training datasets. I present the training details and analyse the performance of these models in section 5.1.

2.1.5 Evaluation Pipeline

The evaluation pipeline of the BabyLM Challenge consists of both text-only and multimodal benchmarks. To evaluate my models, I use the following benchmarks from the BabyLM Challenge:

- BLiMP (The Benchmark of Linguistic Minimal Pairs) (Warstadt et al., 2020) evaluates the linguistic abilities of language models through grammatical acceptability judgements. It consists of minimal pairs of sentences testing a specific phenomenon in syntax, semantics or morphology. Each pair contains one well-formed sentence and one ungrammatical sentence. Models are evaluated by checking whether they assign a higher probability to the grammatical sentence in each pair.
- BLiMP Supplement is a held-out evaluation set introduced in the BabyLM Challenge, consisting of five additional linguistic tasks.
- Elements of World Knowledge (EWoK) (Ivanova et al., 2024) is a zero-shot benchmark that targets specific world concepts such as social interactions, spatial relations and physical dynamics. It uses minimal pairs of context-target combinations, where the same target sentence is plausible given one context but implausible given another. Models are evaluated by checking whether they assign a higher probability to the correct context-target pair.
- Winoground (Thrush et al., 2022) evaluates visio-linguistic compositional reasoning in vision-language models. The dataset consists of hand-curated examples where models must correctly match two images with two captions that contain identical words but in different orders (e.g., "some plants surrounding a lightbulb" vs "a lightbulb surrounding some plants"). Models are evaluated by checking whether they assign a higher probability to the correct caption given the input image.
- VQA v2.0 (Goyal et al., 2017) is an evaluation dataset containing pairs of similar images with identical questions but different correct answers, which forces models to ground their responses in visual content rather than rely on linguistic priors alone. Questions cover multiple categories, such as object recognition, counting and spatial reasoning. Models are evaluated based on which answer they assign the highest probability given the input image and question.

Last year's submissions to the BabyLM Challenge Vision track (Saha et al., 2024; Klerings et al., 2024; AlKhamissi et al., 2024) did not beat the Flamingo and GIT baselines (Hu et al., 2024).

[3] https://huggingface.co/babylm

2.2 Multimodal Artificial Intelligence

Multimodal Artificial Intelligence (AI) aims to replicate the human ability to perceive and integrate information from different sensory channels to form a prediction or decision (Xu et al., 2023; Baltrušaitis et al., 2018). Each sensory ability is generally associated with a specific modality, and a key aspect of human cognition is the brain's capacity to fuse information from distinct modalities into rich representations in order to understand and interact with the world (Zhao et al., 2024).

Research in cognitive sciences suggests that vision contributes to first language acquisition and language understanding. Rose et al. (2009) propose that basic visual cognitive processes in infancy play a role in the acquisition of early language and its expansion between ages one and three. As infants' motor control increases, it dramatically expands their visual access to the world and objects, which, alongside caregiver language, aids in associating words with what they see (Clark and Casillas, 2015). Therefore, in my research, I explore multimodal AI, specifically vision-language models, as computational models of first language acquisition in the context of the BabyLM Challenge.

In this section, I provide an overview of vision-language models (VLMs), focusing on architecture, image tokenisation and embeddings, multimodal fusion and model training.

2.2.1 Core Components of Vision-Language Models (VLMs)

The transformer architecture has become the widely adopted architecture in vision-language models, due to its modality-agnostic nature, flexibility and ability to generalise (Xu et al., 2023). These models treat the textual data as tokens and the visual data as token-like input, which enables them to leverage the attention mechanism of transformers and model complex inner- and cross-modal relationships. Although specific vision-language models (VLMs) vary, most share four principal components (Li et al., 2025):

1. **Vision Encoder:** project raw images (or video frames) into a sequence of embedding features that align with language model embeddings. It is often implemented as a Vision Transformer (ViT) patch encoder, and it is pretrained on rich visual datasets (Dosovitskiy et al., 2020);

2. **Text Encoder:** converts text input into textual embeddings. Early VLMs such as CLIP (Radford et al., 2021), BLIP (Li et al., 2022) and ALIGN (Jia et al., 2021) employ transformer-based text encoders and train both the vision and the text encoders jointly using contrastive learning to align visual and textual representation in a shared latent space (Li et al., 2025). However, more recent models such as LLaVA (Liu et al., 2023) no longer employ a separate text encoder, but simply use a visual encoder with a text decoder;

3. **Cross-attention Mechanism:** dynamically fuses modalities by computing attention scores between image and text tokens;

4. **Text decoder:** generates language outputs conditioned on the fused multimodal features.

2.2.2 Tokenisation of Images and Image Embeddings

To input images into a transformer, they must first be tokenised and projected into the model's embedding space (Li et al., 2025). There exist two common embedding representations:

1. **Global embedding (course-grained):** the entire image is mapped to a single embedding vector representing a global view of the image;
2. **Patch-based embedding (fine-grained):** the image is divided into fixed-size patches (e.g., 16x16) (Dosovitskiy et al., 2020), each of which is then projected by the image encoder to form a token embedding.

2.2.3 Multimodal Fusion

As aforementioned, the attention mechanism is a core component of standard vision-language models, used to fuse the data sourced from different modalities. In the multimodal AI literature, there are three types of fusion defined based on the level at which they occur (Zhao et al., 2024):

1. **Early fusion**: inputs from different modalities are combined together before before being fed to the model;
2. **Intermediate fusion**: features first extracted separately by unimodal encoders are fused and then inputted to the model to generate a prediction/decision;
3. **Late fusion**: separate unimodal models generate independent predictions/decisions, which are then fused in a final step, e.g., weighted average.

Cross-attention, originally introduced in Flamingo (Alayrac et al., 2022), is an example of intermediate fusion and has become one of the primary multimodal fusion approaches (Zhao et al., 2024) over older, more naive fusion operations such as concatenation or addition. Cross-attention follows the same formula as the original self-attention (Vaswani et al., 2017), with the difference that the queries (Q) come from one modality (typically text), while the keys (K) and values (V) originate from the other (image):

$$\text{CrossAttn}\big(Q_{\text{text}},\, K_{\text{img}},\, V_{\text{img}}\big) = \text{Softmax}\left(\frac{Q_{\text{text}}\, K_{\text{img}}^{T}}{\sqrt{d_k}}\right) V_{\text{img}}. \tag{2.1}$$

where $d_{\text{k}} = d_{\text{model}}/n_{\text{heads}}$ is the dimension of each attention head. Alternatively, some state-of-the-art models, including GIT (Wang et al., 2022), Qwen-VL (Bai et al., 2023), DeepSeek-VL (Lu et al., 2024) and Idefics2 (Laurençon et al., 2024b), simply concatenate image tokens with text tokens and apply standard self-attention over the combined sequence.

2.2.4 Pretraining of VLMs

In general, multimodal models are trained using self-supervised objectives (Xu et al., 2023), analogous to language-only models. Moreover, previous work suggests that multi-task training improves

the pretraining of the multimodal transformers (Lu et al., 2020; Xu et al., 2023). Some commonly used losses include masked language modelling, next token prediction, contrastive learning and image-text matching (ITM) (Xu et al., 2023).

Some models adopt a two-stage pretraining regime, where they first pretrain the vision and/or language encoders separately, then end-to-end, e.g., GIT (Wang et al., 2022), ViLBERT (Lu et al., 2019), VL-BERT (Su et al., 2019), while others perform pretraining in one stage e.g., Pixel-BERT (Huang et al., 2020), SimVLM (Wang et al., 2021). Since the advent of Frozen (Tsimpoukelli et al., 2021) and Flamingo (Alayrac et al., 2022), most VLMs build upon large pretrained unimodal backbones (a vision encoder and/or large language model), connecting them via cross-attention or joint self-attention layers rather than training the entire model from scratch (Laurençon et al., 2024a; Koh et al., 2023; Li et al., 2023; Liu et al., 2023).

Chapter 3

Method

3.1 Overview

In this chapter, I present a cognitively-inspired multimodal framework for the BabyLM Vision track that learns language from both ungrounded and visually-grounded text data. My approach differs from existing language-vision models by prioritising cognitive plausibility over raw performance metrics. Specifically, I develop a **dual-stream transformer architecture with three key innovations** that aim to mirror human language processing:

1. **Cognitive alignment through dynamic gating**: Unlike standard vision-language models that use uniform fusion strategies, I implement a token-wise dynamic gating mechanism (Section 3.4) with four variants exploring different granularities and decision levels. This mechanism learns to adaptively weight visual versus linguistic information for each token, aiming to mirror how humans selectively integrate multimodal information. For example, humans may rely more heavily on visual context for concrete words than abstract concepts.

2. **Maximising limited visual information**: Given the constraint of using only a global image embedding during training, I implement multiple strategies to compensate for the limited visual information (Section 3.5). These include modulation techniques that dynamically transform features based on cross-modal context, and a channel attention mechanism to identify salient aspects within the limited visual representation.

3. **Visual grounding via auxiliary objective functions**: I explore two auxiliary objective functions to enhance visual grounding in my framework (Section 3.6): (1) a contrastive learning objective (Radford et al., 2021) which aligns entire captions with images at the sentence level, and (2) LexiContrastive Grounding (Zhuang et al., 2024), which performs word-level alignment between individual tokens and images. While I evaluate my models on general language and multimodal benchmarks (Section 5.3), these auxiliary objectives aim to improve overall language learning by creating stronger associations between linguistic and visual representations.

Moreover, I explore multiple data curriculum strategies (Section 3.7) that could optimise learning in

my proposed framework.

3.2 Motivation

Motivated by cognitive theories of language acquisition and the constraints of the BabyLM Challenge, I propose a novel framework for the BabyLM Vision track. State-of-the-art vision-language models, such as Flamingo (Alayrac et al., 2022) and GIT (Wang et al., 2022), have been designed to predominantly rely on large-scale components pretrained on vast amounts of data (Laurençon et al., 2024a). However, this approach significantly contrasts with the cognitive plausibility and data efficiency goals of BabyLMs.

Previous submissions to the BabyLM Vision track (Saha et al., 2024; Klerings et al., 2024; AlKhamissi et al., 2024) built upon the Flamingo and GIT architectures without significantly modifying them to align with the constraints of the challenge. I hypothesise that these models, originally introduced to be trained on large datasets, lack explicit cognitive motivation and may have performance limitations when scaled down, which leaves substantial room for improvement.

In my framework, I adopt a different approach by making informed architectural and training decisions that optimally utilise the data available, while also being cognitively inspired. Therefore, **my problem statement** is that the training data and number of training epochs are fixed according to the BabyLM Challenge constraints, and the architecture and training regime are variables which I aim to improve.

On the **technical side**, I start with an autoregressive transformer-based architecture that is able to process both text-only and image-caption data, for which I justify my implementation and hyperparameter choices. As features of my model, I explore dynamic gating to improve text and image data fusion and integrate cross-modal modulation strategies in order to enhance feature representation. Moreover, I incorporate contrastive learning objectives at both sentence and word levels motivated by prior work showing that multi-task learning improves the performance of vision-language models (Lu et al., 2020). On the **cognitive side**, I motivate my dynamic gating feature as a mechanism mimicking selective attention, allowing the model to decide how much to rely on linguistic or visual context for each token. This is particularly important for benchmarks like VQA (Goyal et al., 2017), which challenge models to reject both ungrammatical and implausible answers. The cognitively-inspired motivation behind using auxiliary contrastive learning objectives is to improve word acquisition, drawing on research in cognitive science, which shows that visual access to objects and the environment plays a crucial role in early language learning (Clark and Casillas, 2015).

Furthermore, I investigate which training data curriculum strategies best support my framework under the BabyLM Vision track constraints. At a coarse-grained level, I alternate between text-only and image-caption epochs in order to facilitate my choice of auxiliary objectives. At a fine-grained level, I explore how mixing text-only and image-caption data within the same batch, either uniformly or non-uniformly, impacts the training dynamics and generalisation of the model.

Overall, my motivation is based on both cognitive plausibility and computational constraints. The components I introduce in this framework represent a step towards models that learn more like humans do, while contributing to the goals of the BabyLM Vision track and to the broader efforts of building more interpretable and cognitively-plausible language models.

3.3 Base Architecture

I design an autoregressive dual stream transformer as the core model for my framework, drawing inspiration from the architecture of state-of-the-art vision-language models such as LLaVA (Liu et al., 2023) and QWen-VL (Bai et al., 2023). In the following sections, I introduce several features that I build upon my base model aimed at improving both its performance and cognitive plausibility under the constraints of the BabyLM Challenge Vision track. A high-level design of the model is illustrated in figure 3.1.

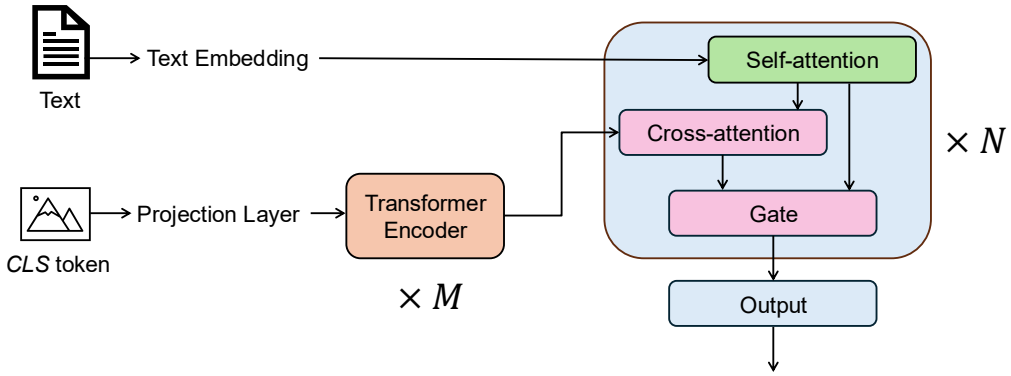


Figure 3.1: Simplified dual-stream architecture. The **text processing stream** (top) embeds text input tokens into a d_{model}-dimensional space and feeds them through an N-layer transformer decoder, which applies masked self-attention, cross-attention to image features and a dynamic gating module to fuse representations. The **image processing stream** (bottom) projects a DINOv2 *CLS* token into the same d_{model}-dimensional space and processes it with an M-layer transformer encoder. Residual connections, normalisation and feed-forward layers are omitted for clarity. The image processing stream, cross-attention and gating modules are skipped for text-only samples.

3.3.1 Core Components

My architecture consists of four main components that enable efficient multimodal learning:

Text Processing Stream. The core of my model is a decoder that processes text input through learned embeddings. I implement a standard text embedding module which combines token embeddings with positional encodings, followed by layer normalisation for training stability. The text stream processes the text data from the text-only dataset as well as the captions from the image-caption dataset.

Image Processing Stream. For the visual input, I use the DINOv2 embeddings provided by the BabyLM Challenge, where each image is encoded as a 768-dimensional *CLS* token. This equates to adding an external pretrained frozen image encoder to my model. A linear layer projects these pretrained visual features into the model's hidden dimensional space.

Despite working with a single global image representation rather than a sequence of patch tokens, I implement a dedicated image encoder after the pretrained frozen DINOv2 one, consisting of transformer encoder layers. Reasons include direct comparison and compatibility with future iterations of this framework using patch-token image embeddings, empirical performance and computational efficiency (compared to alternatives), which I detail in Appendix C.

Multimodal Decoder. The core of my base architecture is a stack of multimodal decoder layers that integrate text and image features through the following three mechanisms applied in sequential order:

1. Multi-head masked self-attention: I apply standard causal self-attention to the text stream. Because my current image input is a single token, it cannot benefit from self-attention. However, future iterations using patch tokens should apply non-causal self-attention to the image input for a richer representation, either in the image encoder or in the multimodal decoder if the image encoder is skipped.

2. Multi-head cross-attention: When image input is available, I perform cross-attention fusion between the text and image features; otherwise, this step is skipped. Cross-attention has been shown to yield superior performance and has become the standard fusion technique in state-of-the-art VLMs (Zhao et al., 2024). The queries originate from the text, while the keys and values are extracted from the image representation.

3. Dynamic gating: Following cross-attention, I apply a dynamic gate that adaptively determines how much to rely on visual versus linguistic information for each token. While cross-attention computes attention weights over the image, determining *what* visual features are relevant, my gating mechanism makes a complementary decision: given those attended features, *how much* should they influence the text representation? I provide a detailed description of the gating mechanism, including other gate variants, in Sections 3.4 and 4.1.

Output Projection. A final layer projects the decoder outputs back to the vocabulary space for next token prediction.

3.4 Dynamic Gating

While dynamic gating in multimodal AI has primarily focused on classification tasks, demonstrating improved robustness and computational efficiency (Xue and Marculescu, 2023; Wang and Wang, 2024; Xie and Zhang, 2020), I ask whether this idea can be repurposed as a cognitively-motivated mechanism for token selection in multimodal autoregressive models.

Research in human cognition found that abstract words primarily activate language-related brain regions, whereas concrete words engage perceptual brain areas (Wang et al., 2010, 2018). Further work (Anderson et al., 2017) showed that functional Magnetic Resonance Imaging patterns for concrete nouns can be decoded by both linguistic and visual representations, but abstract nouns are only decodable via linguistic representations (Wang et al., 2018).

Drawing inspiration from these findings, my hypothesis is as follows: just as human language processing selectively integrates visual information, for example, relying heavily on visual inputs for concrete, perceptual words (e.g., "dog", "red") while defaulting to linguistic knowledge for abstract terms (e.g., "therefore", "impossible"), a token-wise dynamic gating mechanism could teach a model to make similar fine-grained fusion decisions. By conditioning each gate on both the current text hidden state and the cross-attention features, the model can learn to amplify the vision input when it truly informs the next word and ignore it when it does not. This approach contrasts with Flamingo's gated cross-attention dense blocks, which apply uniform layer-wise gating parameters across all tokens, and are thus unable to adapt based on individual token semantics.

I implement four variants of a dynamic gating mechanism, varying along two axes: **(1) granularity**, whether the gate is computed per feature or per token, and **(2) soft vs hard**, whether the gate outputs continuous weights or discrete decisions. The granularity axis investigates whether different tokens require different subsets of visual features (e.g., colour features for "red", spatial features for "above") or whether coarse per-token gating is sufficient for effective vision-language fusion. The soft vs hard gating axis examines whether binary selection (fully using or discarding features) or continuous weighting of features yields more interpretable fusion patterns and better performance. Figure 3.2 illustrates the conceptual output for each type of gate.

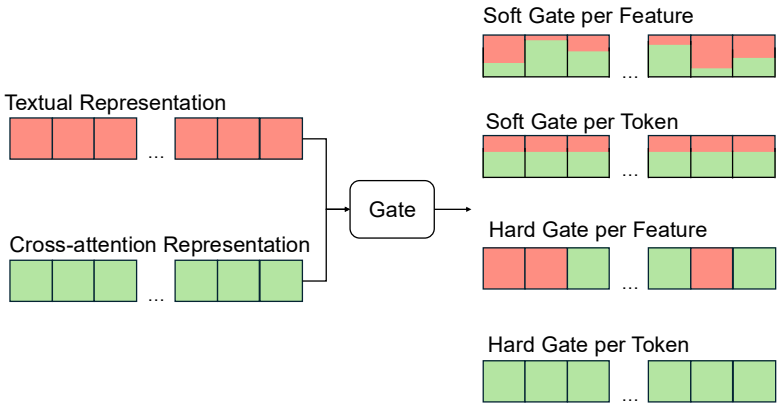


Figure 3.2: Conceptual output of different gating strategies for fusing textual and cross-attention representations. Each rectangular box represents a token, with the cells within representing dimensions. Red represents textual features, green represents cross-attention features and mixed colours represent fused features. Soft gates apply continuous weights, while hard gates make binary decisions, either per-feature (each dimension independently) or per-token (all dimensions together).

The technical implementation details for the four gating variants are available in Section 4.1.

Wang et al. (2018) asked a similar question about how to dynamically weigh linguistic and visual input based on word type. However, their goal was to create static word embeddings relying on weak supervision. In contrast, I propose using dynamic gating as an unsupervised mechanism during autoregressive generation, where the model decides at each step which modality should produce the next token.

3.5 Feature Representation

The BabyLM Challenge constraint of using only a global *CLS* token, while computationally efficient, limits the spatial visual information available to the model. Traditional vision-language models benefit from patch token representations that preserve spatial information and enable fine-grained visual grounding. Therefore, the next aspects I investigate in my framework are methods of maximising the utility of the *CLS* token. I explore two complementary modulation techniques, FiLM (Perez et al., 2018) and DyIntra (Gao et al., 2019), which dynamically reshape one set of features based on another, as well as a global channel-attention enhancement. These approaches target different aspects of the representation bottleneck: modulation techniques address cross-modal feature interaction, while channel attention addresses intra-modal feature refinement. I evaluate these methods at several integration points within my architecture to determine which approach most effectively compensates for the lack of spatial visual information.

Motivation for modulation. While my dynamic gating mechanism determines *how much* information to incorporate from each modality, FiLM and DyIntra determine *how* that information should be transformed. I hypothesise that this complementary component to my framework allows the model to learn richer fusion strategies.

Motivation for channel attention. I investigate whether sharpening some of the signals in the global *CLS* embedding can enhance its utility. Therefore, I implement a channel attention mechanism to determine *what* is meaningful within the image features.

The technical details of the feature modulation and channel attention enhancements I implement in my framework are available in Section 4.2.

3.6 Objective Functions

As previous work in vision-language models suggests (Lu et al., 2020), a multi-task objective can improve the model's performance. Therefore, in my framework, I explore training my models using two auxiliary functions, Contrastive Language-Image Pre-training (CLIP) (Radford et al., 2021) and LexiContrastive Grounding (LCG) (Zhuang et al., 2024). Both functions aim to ground textual representations in visual concepts through contrastive learning, creating a shared embedding space where semantically related image-text pairs are positioned closer together. However, they operate at differ-

ent levels of granularity: CLIP aligns entire captions with their corresponding images at the sentence level, while LCG performs alignment at the word level between individual tokens and images.

My choice for these auxiliary functions is cognitively-motivated, as recent research shows that visual grounding at both sentence and word levels can improve word acquisition in low-data regimes (Zhuang et al., 2023). A CLIP objective could capture global associations that support contextual understanding, while an LCG objective might reflect fine-grained grounded learning. By investigating these objectives, I explore whether an explicit contrastive mechanism can enhance language learning under the constraints of the BabyLM Challenge. However, as I discuss in my results (Section 5.3), the effectiveness of these auxiliary objectives significantly depends on various factors, including the visual representation format, batch size and training data.

The technical details of each auxiliary objective function as implemented in my framework are available in Section 4.3.

3.7 Data Curriculum

Since the training dataset consists of both text-only data and image-caption data, each accounting for 50M words, I implement and analyse the following data curriculum strategies in my training:

- **Coarse-grained epochs**: I load the text-only and image-caption data in separate PyTorch (Imambi et al., 2021) data loaders, where each data loader alone is used for one epoch. For the 10 epochs constraint of the BabyLM Challenge, this results in 10 text-only epochs and 10 image-caption epochs. I then experiment with the following:
    1. Alternating between image-caption epochs and text-only epochs;
    2. Training on all text-only epochs first, then on the image-caption epochs;
    3. Training on all image-caption epochs first, then on the text-only epochs.
- **Fine-grained epochs**: For the fine-grained epochs, I define the following two training strategies:
    1. I load both the text-only data and the image-caption data in the same data loader, where I pair the text-only data with image tensors filled with 0s for uniformity. The cross-modality path is still skipped in the text-only samples. The original text data is provided in *.txt* files, and I process each text line as one sample. For the image-caption data, I process each (image, caption) pair as one sample. In this setting, the text-only data has twice as many samples as the image-caption data. Therefore, loading and shuffling them in the same data loader results in a non-uniform distribution between the two and more unstable training.
    2. For a uniform distribution between the image-caption data and the text-only data, I take inspiration from the GitHub repository [1] used to train the BabyLM 2024 Challenge base-

[1] https://github.com/aaronmueller/babylm_multimodal_training

lines, where the authors pair each text-only input with one image-caption input in the same batch sample, resulting in uniform batches. Therefore, in one training step, I perform two forward passes: one using the text-only input and one using the image-caption input. I then sum the losses from each pass and do one backward propagation using the total loss. However, since there are twice as many text-only samples than image-caption samples, this results in training the model twice on the image-caption dataset. For 10 training epochs, this equals 10 text-only epochs and 20 image-caption epochs.

For a fair comparison among all of the methods I implement in this work, I alternate between text-only epochs and image-caption epochs in my experiments exploring architectural changes and auxiliary objective functions. That is because the contrastive learning objective functions compute similarity scores between a caption and all the images in a batch. If the batch contains (many) text-only samples, it cancels the effect of the auxiliary losses. Moreover, the empirical results I obtain in section 5.4 support this choice among the data curriculum strategies I define.

Chapter 4

Design and Implementation

In this chapter, I present the technical implementation details for the three components I explore in my multimodal framework: dynamic gating (Section 4.1), feature representation enhancements (Section 4.2) and auxiliary objective functions (Section 4.3). In Section 4.4, I provide details about the implementation of my models, training setup and data preprocessing, followed by the evaluation pipeline in Section 4.5.

4.1 Dynamic Gating

For my framework, I define four dynamic gating variants that operate at different granularity and decision levels: *soft gate per feature*, *soft gate per token*, *hard gate per feature*, *hard gate per token*.

Input and Output. All four versions of the dynamic gate have the same input and output. Let $h_{\text{text}} \in \mathbb{R}^{B\times T\times d_{\text{model}}}$ be the text hidden states after self-attention and $h_{\text{crossAttn}} \in \mathbb{R}^{B\times T\times d_{\text{model}}}$ be the output of the cross-attention between text and image, where B is the batch size and T is the sequence length. Then, the input to the dynamic gate is the concatenation of two hidden representations, $[h_{\text{text}}; h_{\text{crossAttn}}] \in \mathbb{R}^{B\times T\times 2d_{\text{model}}}$. The output is represented by $h_{\text{fused}} \in \mathbb{R}^{B\times T\times d_{\text{model}}}$, which combined the pure linguistic representation with the visually-enriched representation based on the gating weights. In the case of a text-only input to the model, the dynamic gate module is skipped, and h_{text} flows directly through the residual connection.

4.1.1 Soft Gate per Feature

This variant computes a continuous weight for each feature dimension $i \in \{0, ..., d_{\text{model}} - 1\}$ using the sigmoid function. Concretely, the gate vector is computed as:

$$g = \sigma\big(\text{Linear}\,[\,h_{\text{text}};\; h_{\text{crossAttn}}\,]\big) \;\in\; [0,1]^{B\times T\times d_{\text{model}}} \tag{4.1}$$

The dynamically fused representation is then calculated as:

$$h_{\text{fused}} = g \odot h_{\text{text}} + (1 - g) \odot h_{\text{crossAttn}} \tag{4.2}$$

I use this variant of dynamic gating in the base model.

4.1.2 Soft Gate per Token

The soft gate per token calculates a single continuous weight (a scalar), which I apply to all features in the hidden representations:

$$\begin{aligned} g &= \sigma\Big(\text{Linear}\big([\, h_{\text{text}};\, h_{\text{crossAttn}}]\big)\Big) \;\in\; [0,1]^{B\times T\times 1}, \\ h_{\text{fused}} &= g \odot h_{\text{text}} + \big(1 - g\big) \odot h_{\text{crossAttn}} \end{aligned} \tag{4.3}$$

4.1.3 Hard Gate per Feature

Drawing inspiration from Xue and Marculescu (2023), I extend the soft gating variants to a hard selection mechanism using the Gumble-Softmax reparametrisation trick (Jang et al., 2016).

The hard gate per feature variant enforces each dimension to choose completely between linguistic or visually-enriched representations. I first compute a 2-way discrete choice for the two hidden representations using a linear layer:

$$g = \text{Linear}\left([\, h_{\text{text}};\, h_{\text{crossAttn}}\,]\right) \;\in\; [0,1]^{B\times T\times d_{\text{model}}\times 2} \tag{4.4}$$

Each pair of logits $(l_{b,t,i,0}, l_{b,t,i,1})$ corresponds to the scores for "use h_{text}" versus "use $h_{\text{crossAttn}}$" for feature i at position (b,t). A straightforward hard gate would then be:

$$g_{b,t,i} = \arg\max\big(l_{b,t,i,0},\, l_{b,t,i,1}\big) \tag{4.5}$$

However, since g is a one-hot vector, it is not differentiable. Therefore I employ a soft gate $\tilde{g}$ during training using Gumble-Softmax, similar to Xue and Marculescu (2023), to enable back-propagation:

$$\tilde{l}_{b,t,i,j} = \frac{l_{b,t,i,j} + z_{b,t,i,j}}{\tau}, \quad z_{b,t,i,j} \sim \text{Gumbel}(0,1) \tag{4.6}$$

where τ is the Softmax temperature. I then apply Softmax over the two classes and select the probability corresponding to h_{text} as the soft gate:

$$y_{b,t,i,j} = \frac{\exp\big(\tilde{l}_{b,t,i,j}\big)}{\sum_{k=0}^{1}\exp\big(\tilde{l}_{b,t,i,k}\big)}, \quad y \;\in\; [0,1]^{B\times T\times d_{\text{model}}\times 2} \tag{4.7}$$

$$\tilde{g}_{b,t,i} = y_{b,t,i,0}, \quad \tilde{g} \;\in\; [0,1]^{B\times T\times d_{\text{model}}} \tag{4.8}$$

h_{fused} is then be computed as:

$$h_{\text{fused}} = \tilde{g} \odot h_{\text{text}} + \big(1 - \tilde{g}\big) \odot h_{\text{crossAttn}} \tag{4.9}$$

During training, I anneal the Softmax temperature τ from 1.0 to 0.1 over 80% of the training steps of an image-caption epoch, gradually transitioning from soft to nearly discrete selection. During inference, I convert $\tilde{g}$ to a true one-hot gate g using $\arg\max$.

4.1.4 Hard Gate per Token

In the per token variant of the hard gate, I collapse the feature-wise gate into a single binary decision. The calculations, training and inference remain the same as in subsection 4.1.3, yet the shape of the parameters changes. The summary of the calculations in this variant is as follows:

$$l = \text{Linear}\big([\, h_{\text{text}};\, h_{\text{crossAttn}}]\big) \;\in\; \mathbb{R}^{B\times T\times 2} \tag{4.10}$$

$$y = \text{GumbelSoftmax}\big(l,\, \tau\big) \;\in\; \mathbb{R}^{B\times T\times 2} \tag{4.11}$$

$$\tilde{g}_{b,t} = y_{b,t,0} \;\in\; [0, 1] \tag{4.12}$$

$$h_{\text{fused}} = \tilde{g} \odot h_{\text{text}} + \big(1 - \tilde{g}\big) \odot h_{\text{crossAttn}} \tag{4.13}$$

where the scalar $\tilde{g}$ is broadcasted over all d_{model} features.

4.2 Feature Representation

4.2.1 Feature-wise Linear Modulation (FiLM)

To address the limited representational capacity of a single *CLS* token, I incorporate Feature-wise Linear Modulation (FiLM) (Perez et al., 2018) as an intra-modal conditioning mechanism. FiLM modulates neural network features through a feature-wise affine transformation, enabling one modality or context to dynamically influence another. Specifically, it applies scaling and shifting to a feature map based on a conditioning input, and can be easily implemented in transformers as follows:

Let $h_{m_1}, h_{m_2} \in \mathbb{R}^{B\times T\times d_{\text{model}}}$ be hidden state feature representations with m_1 indicating the primary features and m_2 the conditioning features, $m_1 \neq m_2$. Then,

$$\text{FiLM}\big(h_{m_1},\, h_{m_2}\big) = \gamma \odot h_{m_1} + \beta \tag{4.14}$$

where $\gamma, \beta \in \mathbb{R}^{B\times T\times d_{\text{model}}}$ are scaling and shifting parameters predicted by linear layers from h_{m_2}.

4.2.2 Dynamic Intra Modulation (DyIntra)

Alternatively to FiLM, I explore the DyIntra module proposed in (Gao et al., 2019), a scaling mechanism that modulates primary features using conditioning features via a simple gating mask. DyIntra predicts a positive-only gain for each representation, allowing it to boost its own hidden features based on cross-modal context without shifting.

Formally, let $h_{m_1}, h_{m_2} \in \mathbb{R}^{B\times T\times d_{\text{model}}}$ be hidden state feature representations, $m_1 \neq m_2$. Then, DyIntra computes:

$$m = \sigma\big(\text{Linear}(m_2)\big) \in [0,1]^{B\times T\times d_{\text{model}}} \tag{4.15}$$

$$\text{DyIntra}\big(h_{m_1}, h_{m_2}\big) = (1+m) \odot h_{m_1} \tag{4.16}$$

Choosing m_1 and m_2. There are several points in my base model where I could integrate a FiLM or DyIntra modulation module. I evaluate and motivate three such choices as follows:

1. m_1 = self-attention (output), m_2 = image: modulating the text self-attention output with visual features may allow the model to adjust how text tokens relate to each other based on visual context;
2. m_1=cross-attention (output), m_2 = image: modulating cross-attention features with the original image may refine the vision-language fusion by emphasising features that align with the global visual representation;
3. m_1 = image, m_2 = text: modulating image features based on textual context may allow the model to dynamically highlight relevant visual information for the current linguistic processing needs.

4.2.3 Channel Attention

To implement channel attention for only one image token, I use the Excitation formula from the Squeeze-and-Excitation method (Hu et al., 2018) as follows:

$$h'_{\text{image}} = \sigma\Big(W_2\,\text{ReLU}\big(W_1\,h_{\text{image}}\big)\Big) \odot h_{\text{image}}, \quad W_1 \in \mathbb{R}^{(d_{\text{model}}/r\times d_{\text{model}})}, \quad W_2 \in \mathbb{R}^{(d_{\text{model}}\times d_{\text{model}}/r)} \tag{4.17}$$

where h_{image} is the output of the image encoder and $r = 16$ is the reduction ratio. I expect this method to help the model focus on the most informative features of the visual embedding, improving the quality of image representations.

4.3 Objective Functions

4.3.1 Contrastive Language-Image Pre-training (CLIP)

I incorporate CLIP's objective function into the training of my base model for image-caption epochs steps, as follows:

For each sample in a batch, I first extract pooled representations from both image and text modalities. For text, I compute mean pooling over the the output of the text embedding module, which I denote t_{pooled}. For the image, I use the output of the image encoder directly as its length is 1, i_{pooled}. Both t_{pooled} and i_{pooled} are then projected to a shared contrastive embedding space through specific linear projection layers. I L2-normalise both representations before computing similarity scores. The contrastive loss is formulated as a bidirectional InfoNCE objective (Oord et al., 2018) with learnable temperature τ. It combines text-to-image and image-to-text matching losses, where each direction maximises the similarity between matched pairs while minimising similarity with all other pairs in the batch. The final loss is computed as $\mathcal{L}_{\text{contrastive}} = \frac{1}{2}(\mathcal{L}_{\text{t2i}} + \mathcal{L}_{\text{i2t}})$.

The complete training objective then becomes:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NTP}} + \lambda \mathcal{L}_{\text{contrastive}} \tag{4.18}$$

where λ represents the weight of the contrastive loss and NTP stands for *next-token prediction*.

In my experiments, I initialise τ to 0.07 and constraint it between 0.05 and 1 during training for stability, and set λ to 1.

4.3.2 LexiContrastive Grounding (LCG)

LexiContrastive Grounding (LCG) (Zhuang et al., 2024) is a training procedure that implements a grounded language learning objective similar to CLIP. While CLIP operates at sentence level, LCG computes similarity scores at the word level. To calculate the cross-modality contrastive learning loss, LCG leverages the first hidden layer of a model, which stores lexical information. The authors also limit the attention mask applied to the first layer to a previous two-word window in order to encode less linguistic context. The contrastive loss is then calculated per batch from all the token-level representations outputted by the first layer.

For my model, I adapt and implement the LCG during the image-caption epochs as follows:

Let $(\text{text}_i, \text{image}_i)$ represent the image-caption pairs in a batch, where $i \in \{1, 2, \ldots, n\}$ and n is the batch size. Each caption text_i contains m_i tokens: $(t_{i,1}, t_{i,2}, \ldots, t_{i,m_i})$.

To obtain lexically-focused representations, I extract the textual representation from the first layer after the residual connection applied to self-attention:

$$h_1\big(\text{text}_i\big) = \text{text}_i + \text{SelfAttn}\big(\text{LayerNorm}(\text{text}_i)\big) \tag{4.19}$$

In my implementation, I experimented with applying a narrow two-word attention mask, however, I noticed conflicts with the next token prediction loss. Specifically, applying the two-word attention mask in the first layer was preventing the next token prediction loss from decreasing. I tried applying the two-word attention mask solely to extract the hidden representation, then switching to the original causal mask for the rest of first layer's forward pass, as well as skipping the cross-modal fusion in

the first layer, but neither approach fixed the problem. Therefore, I decided to use the standard causal attention mask when extracting the first layer textual hidden representation, as ablation studies in the original research (Zhuang et al., 2024) did not indicate a significant loss in performance for this case.

Let $h_1(\text{text}_i, j) \in \mathbb{R}^{d_{\text{model}}}$ be the first layer representation of the j-th token (the j-th row of the matrix) in the i-th caption and $\text{enc}(\text{image}_i) \in \mathbb{R}^{d_{\text{model}}}$ represent the output of my image encoder for the i-th image. Then, the matching score between the j-th token in the caption k and image i is calculated as:

$$s(i, j, k) = \frac{\left(M_{\text{image}} \cdot \text{enc}(\text{image}_i)\right)^T \cdot \left(M_{\text{text}} \cdot h_1(\text{text}_k, j)\right)}{\tau} \tag{4.20}$$

where $M_{\text{image}}, M_{\text{text}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ are learned projection matrices and τ is a learnable temperature parameter, which I clamp between $[0.05, 2.0]$ for training stability.

For each valid token position, I then compute the LCG contrastive learning loss as:

$$\mathcal{L}_{\text{LCG}} = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \mathbf{1}_{\text{valid}}(i, j) \cdot \frac{1}{2}\left[\ell_1(i, j) + \ell_2(i, j)\right] \tag{4.21}$$

where $\mathbf{1}_{\text{valid}}(i, j)$ is an indicator function for non-padded tokens, and:

$$\ell_1(i, j) = -\log \frac{e^{s(i,j,i)}}{\sum_{k=1}^{n} e^{s(k,j,i)}}, \quad \ell_2(i, j) = -\log \frac{e^{s(i,j,i)}}{\text{neg}(i, j)}. \tag{4.22}$$

The negative term $\text{neg}(i, j)$ is defined as:

$$\text{neg}(i, j) = e^{s(i,j,i)} + \sum_{\substack{k=1 \\ k \neq i}}^{n} \sum_{o=1}^{m_k} \mathbf{1}_{\text{valid}}(k, o) \cdot e^{s(i,o,k)} \tag{4.23}$$

The total loss combines next-token prediction with word-level contrastive learning loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{NTP}} + \lambda \cdot \mathcal{L}_{\text{LCG}} \tag{4.24}$$

where λ is a hyperparameter controlling the strength of visual grounding. I set λ to 0.3 through trial and error such that $\mathcal{L}_{\text{NTP}}$ and $\mathcal{L}_{\text{LCG}}$ have the same magnitude.

I use auxiliary functions only during the image-caption epochs, as the image processing stream is skipped for text-only samples.

4.4 Experiments Setup

For all architectural features and training strategies I define in subsections 3.4-3.7 and 4.1-4.3, I conduct experiments in the form of ablation studies in order to evaluate each potential improvement in isolation. I select my base architecture, described in section 3.3, and define one experiment per feature. I train each enhanced model in the same conditions and evaluate it on the BabyLM Challenge

benchmarks: BLiMP, BLiMP Supplement, EwoK, Winoground and VQA.

I describe the base model implementation details in subsection 4.4.1 and the training regime I use in Subsection 4.4.2. A summary of all the experiments I define is available in Appendix B.

4.4.1 Base Model Implementation Details

I implement the dual stream transformer in PyTorch (Imambi et al., 2021), following the architecture I introduce in section 3.3. I summarise my hyperparameter choices for the base model in Appendix A.

I use pre-layer normalisation rather than post-layer normalisation in my implementation as previous research shows that pre-layer normalisation provides better training stability for networks larger than six layers (Takase et al., 2022), which is crucial given my limited training budget and inability to perform extensive hyperparameter searches.

Following standard transformer design, I use residual connections around each sub-layer (feed-forward networks, self-attention and cross-attention).

While the image encoder may be over-parameterised for single token processing, empirical results validate this choice (Appendix C), and it ensures architectural consistency and directly comparable results for future extensions to patch-based visual inputs.

4.4.2 Training Details

I train all of my models using the hyperparameters summarised in table 4.1, with the exception of a few changes for the auxiliary objective function and data curriculum experiments. In the case of the auxiliary objective function experiments, I increase the batch size from 64 to 128 as a larger batch size is recommended for contrastive learning (Chen et al., 2020), which results in a total of 553,510 steps. Due to computational constraints, I was not able to select a larger batch size. In the case of the data curriculum experiments, the data order differs according to the strategy I define for that experiment. For the model trained using LexiContrastive Grounding as the auxiliary function, I use weight tying as recommended in the original work (Zhuang et al., 2024).

I select a learning rate of 5e-5 to ensure training stability, despite this being conservative for the model size. While alternating between text-only and image-caption epochs improves performance on my benchmarks (as shown in section 5.4), this training regime can cause gradient instability when transitioning between epoch types. Therefore, I adopt a lower learning rate to mitigate this risk.

4.4.3 Data Pipeline Details

The text-only training dataset is provided in *.txt* files, while the multimodal one is provided in *.json* files for the captions and *.npy* for the image embeddings, where each row in the numpy array embeds one image as a *CLS* token of dimension 768. I load the text-only data as one training sample per line,

Training Hyperparameter	Value
Data order	Alternating between text-only and image-caption epochs
Number of epochs	10 text-only and 10 image-caption
Total number of steps	1,107,020
Checkpoints saved	Every 50,000 steps
Batch size	64
Learning rate	5e-5
Learning rate schedule	Cosine annealing
Optimiser	AdamW with $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e-8$
Number of steps for warmup	$\sim$1%
Weight decay	0.01
Gradient clipping norm	1.0
Main loss function	Cross-entropy
Random seed	42

Table 4.1: The hyperparameters list for my base training regime.

and the image-caption data as one (image, caption) pair representing one training sample. I do not perform any preprocessing on either data.

For the text-only data and the captions, I tokenise the text using the GPT-2 tokeniser (Radford et al., 2019), as my model is autoregressive. I also add the *BOS* and *EOS* special tokens at the beginning and end of each text-only/caption sample, respectively.

I split the data into 80% training, 10% validation and 10% held-out test sets. In order to ensure that all models are trained on the same data, I save the data split indices and reuse them for all experiments. I shuffle the training dataset independently for each run while maintaining consistent train/validation/test partitions.

4.5 Evaluation Pipeline

4.5.1 Performance Scores

To evaluate my framework, I select five of the benchmarks proposed in the BabyLM Challenge: BLiMP and BLiMP Supplement for grammar, EWoK for world knowledge, Winoground for vision-linguistic compositional reasoning and VQA for image-based question answering. I provide a detailed description for each benchmark in subsection 2.1.5.

I follow the same approach as the BabyLM Challenge 2024 evaluation pipeline[1], using the *lm-harness* library[2], which provides part of the evaluation code for the benchmarks. I then implement a wrapper class for my model that returns the log-probabilities for the input. More specifically, the input for each benchmark is of the form

$$\big(\text{Optional}(\text{context}),\ \text{continuation},\ \text{Optional}(\text{image})\big) \tag{4.25}$$

[1] https://github.com/babylm/evaluation-pipeline-2024
[2] https://github.com/EleutherAI/lm-evaluation-harness

and the calculation for each benchmark is as follows:

For any given example, I first prepend a *BOS* token and concatenate the context (if present) and continuation, then tokenise to obtain a sequence of tokens $\{t_1, t_2, \ldots, t_N\}$. I use the *BOS* token as I train my model using *BOS/EOS* tokens. However, I do not add or count the *EOS* in the performance score. I perform one forward pass over the tokens and retrieve the logits $\ell_i(v)$ over the vocabulary at each position i, with no temperature scaling. I convert logits to probabilities by

$$p\big(t_i \mid t_{<i},\, \text{Optional(image)}\big) = \frac{\exp\big(\ell_i(t_i)\big)}{\sum_v \exp\big(\ell_i(v)\big)} \tag{4.26}$$

and define the total log-likelihood of the sequence as

$$\log \text{likelihood}(t_{1:N}) = \sum_{i=1}^{N} \log p\big(t_i \mid t_{<i},\, \text{Optional(image)}\big). \tag{4.27}$$

- **BLiMP/BLiMP Supplement:** There are no *context* or *image* inputs. Each test consists of a minimal-pair $\big(s_{\text{gram}},\, s_{\text{ungram}}\big)$ regarded as *continuation*. I compute

$$\log \text{likelihood}\big(s_{\text{gram}}\big) \quad \text{and} \quad \log \text{likelihood}\big(s_{\text{ungram}}\big), \tag{4.28}$$

  and count the pair as correct if $\log \text{likelihood}(s_{\text{gram}}) > \log \text{likelihood}(s_{\text{ungram}})$. The reported score is

$$\frac{\#\{\text{correct pairs}\}}{\#\{\text{total pairs}\}} \tag{4.29}$$

- **EWoK:** Each example consists of a *context*, for which the model is provided one correct and one incorrect *continuation*. For each continuation c, I compute

$$\log \text{likelihood}(c) = \sum_{i=1}^{|c|} \log p\big(c_i \mid \text{context},\, c_{<i}\big) \tag{4.30}$$

  The reported score is the fraction of prompts for which the gold continuation has the higher log-likelihood.

- **Winoground:** Each example consists of one image i and two captions $\big(c_{\text{correct}}, c_{\text{incorrect}}\big)$. There is no *context* and the captions are regarded as *continuation*. I compute two log-likelihoods,

$$\log \text{likelihood}\big(c_{\text{correct}} \mid i\big) \quad \text{and} \quad \log \text{likelihood}\big(c_{\text{incorrect}} \mid i\big) \tag{4.31}$$

  The reported score is the fraction of examples for which

$$\log \text{likelihood}\big(c_{\text{correct}} \mid i\big) > \log \text{likelihood}\big(c_{\text{incorrect}} \mid i\big) \tag{4.32}$$

- **VQA:** Each example is a *(question, multiple answer choices, image)* tuple, where the *question* is the *context* and the *answer* is the *continuation*. Each question example comes with one correct answer and seven incorrect answers $(a_0, a_2, \ldots, a_7)$. Therefore, for question q, answer options $(a_0, a_2, \ldots, a_7)$ and corresponding image i, I calculate eight probabilities

$$\log \text{likelihood}\big(a_j \mid q, i\big), \quad j \in \{0, ..., 7\} \tag{4.33}$$

  The reported score is the fraction of tuples *(question, multiple answer choices, image)* for which the correct answer has the highest log-likelikhood.

4.5.2 Gate Selection

To understand whether there is a cognitive link to the dynamic gating mechanism of my framework, I investigate the correlation between the gate weight for next token prediction and part-of-speech, concreteness, imageability, familiarity and age of acquisition.

To achieve this, I use the portion of the Localized Narratives image-caption dataset that I save as a held-out test set and select samples accounting for a total of 1,034 tokens. For each *(image, caption)* input, I perform one forward pass through my trained model and extract the gate weights from the final decoder layer. For the *per feature* gates, I average the per feature scores to obtain the final gate weight. Since the gate at position $t-1$ is used to predict the token at position t, I align each predicted word with its corresponding gate value from the previous position. I then augment each word with its part-of-speech tag using spaCy[3] and retrieve psycholinguistic metrics (age of acquisition, imageability, concreteness and familiarity) from the MRC Psycholinguistic Database (Coltheart, 1981) using the version available on HuggingFace[4] which is extracted from the online database[5]. This results in tuples of type *(word, gate weight, metric score)*, which I use to investigate whether the learned gating mechanism exhibits systematic relationships with these cognitive and linguistic properties.

[3] https://github.com/explosion/spaCy
[4] https://huggingface.co/datasets/StephanAkkerman/MRC-psycholinguistic-database
[5] https://websites.psychology.uwa.edu.au/school/mrcdatabase/uwa_mrc.htm

Chapter 5

Results

In this chapter, I present the experimental results of my cognitively-inspired multimodal framework. I first provide an overview of my base model's performance compared to the BabyLM Challenge 2024 multimodal baselines (Section 5.1), followed by performance analysis of the dynamic gating mechanisms (Subsection 5.2.1), feature enhancement techniques (Subsection 5.2.2) and auxiliary objective functions (Section 5.3). I then discuss the effect of the defined data curriculum strategies (Section 5.4) and resulting training dynamics (Section 5.5). Finally, I investigate the interpretability of learned gating patterns (Section 5.6) to assess whether there is a link between the gating mechanisms in my framework and human cognition.

5.1 Baselines

Model	**BLiMP**	**BLiMP Supplement**	**EWoK**	**Winoground**	**VQA**
Base Model	**75.53 ± 0.16%**	55.71 ± 0.57%	50.41 ± 0.57%	51.74 ± 1.83%	50.02 ± 0.31%
Flamingo	70.88 ± 0.16%	**65.02 ± 0.54%**	**52.7 ± 0.57%**	51.6 ± 1.83%	52.3 ± 0.31%
GIT	65.35 ± 0.17%	62.69 ± 0.54%	52.41 ± 0.57%	**55.5 ± 1.82%**	**54.1 ± 0.31%**

Table 5.1: The performance of my base model compared to the Flamingo and GIT BabyLM Challenge 2024 baselines on BLiMP, BLiMP Supplement, EWoK, Winoground and VQA.

Table 5.1 provides a reference for my base model's performance compared to the Flamingo and GIT baselines from the BabyLM Challenge 2024 on the five selected evaluation benchmarks: BLiMP, BLiMP Supplement, EWoK, Winoground and VQA. At the time of writing, there are no publicly available baselines for the BabyLM Challenge 2025 Vision track. The 2024 baselines were trained using a different regime than the constraints introduced in 2025. Therefore, while these baselines provide useful context for interpreting performance, direct comparisons are limited by the different training constraints. Specifically, the organisers trained the 2024 Flamingo and GIT baselines using patch token representation for the image, despite providing only global embeddings for the challenge participants. Moreover, the baseline models were trained on 20 epochs worth of text-only data and 40

epochs worth of image-caption data. This contrasts with the 10-epoch constraint for the 2025 challenge, which I respect in this work. The learning rate also differs. The Flamingo and GIT baselines were trained using a learning rate of 1e-4, while I used a learning rate of 5e-5 for training stability.

As shown, my base model achieves a significantly higher score on BLiMP (over 10% higher than GIT and almost 5% higher than Flamingo) and competitive scores for EWoK, Winoground and VQA. I suggest that my base model performs better than Flamingo and GIT on BLiMP due to architectural differences. Even when I train a variant of my model under a similar regime to the 2024 BabyLM Challenge baselines, I observe similarly superior results on BLiMP (see Section 5.4). One reason for these results could be the clear separation between the text stream and image stream, as well as consistent fusion in my base model, which contrasts with Flamingo and GIT. In particular, in my proposed model, the self-attention module in the first decoder layer processes only textual input, whereas GIT concatenates the image token(s) with the text input before they are fed into the model, which could introduce noise when extracting linguistic signals. Furthermore, my model consistently injects visual features at every decoding layer, while Flamingo introduces visual information every few layers. This sporadic fusion could limit the model's ability to learn a stable textual representation.

One main factor for my base model's lower performance on BLiMP Supplement could be the training regime I implement, the alternation between text-only and image-caption epochs. In Section 5.5, I analyse this training regime and find that the text-only dataset supports the BLiMP Supplement benchmark far less than the image-caption one, which may introduce performance instability. Another reason for the lower performance on BLiMP Supplement could be the number of training epochs, as the BabyLM Challenge 2024 baselines were trained on twice as much text-only data and four times as much image-caption data.

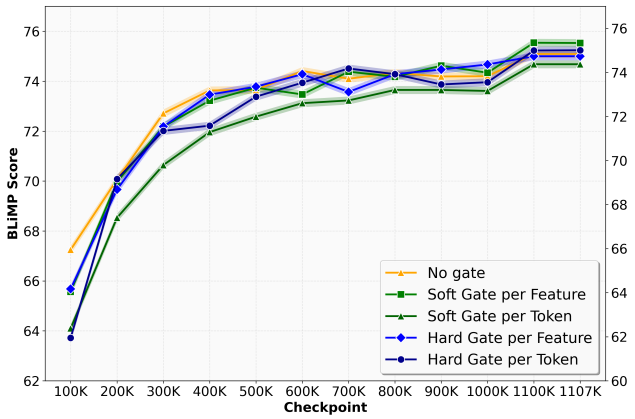
Despite the disparity in the number of training epochs, my model achieves competitive performance on EWoK, Winoground and VQA, underscoring the effectiveness of my proposed framework.

5.2 Performance of Architectural Features

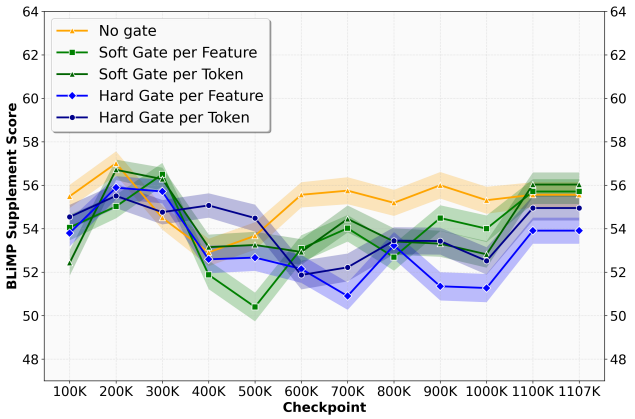
5.2.1 Dynamic Gating

In figure 5.1, I analyse the performance of my base model with different gate variants, as well with no gate module, every 100,000 steps on the BLiMP, BLiMP Supplement, EWoK, Winoground and VQA benchmarks. The goal for the gate module is to maintain the model's performance on text-only benchmarks while increasing its scores on multimodal benchmarks by allowing it to choose how much to rely on previous text versus the input image.

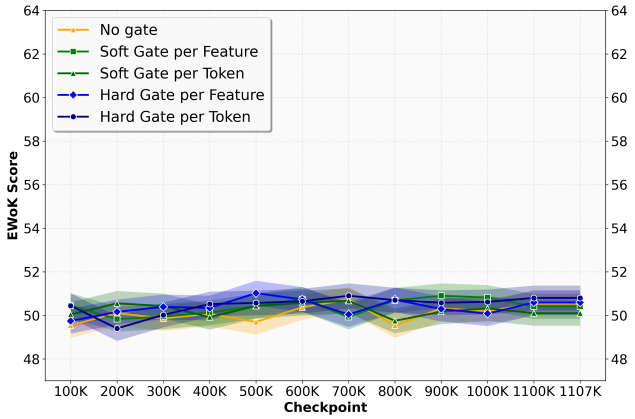
BLiMP. In subfigure 5.1a, it can be seen that all models incorporating different gate variants follow the same pattern when evaluated on BLiMP every 100,000 steps. The *no gate* variant achieves a higher score in early training by 1.5%-3%. However, this gap with the other models closes over the rest of the training. The *per token* gate variants seem to have slightly worse performance than the



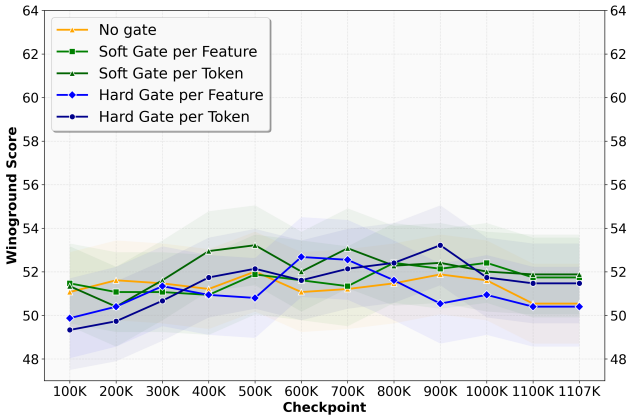
(a) BLiMP scores.



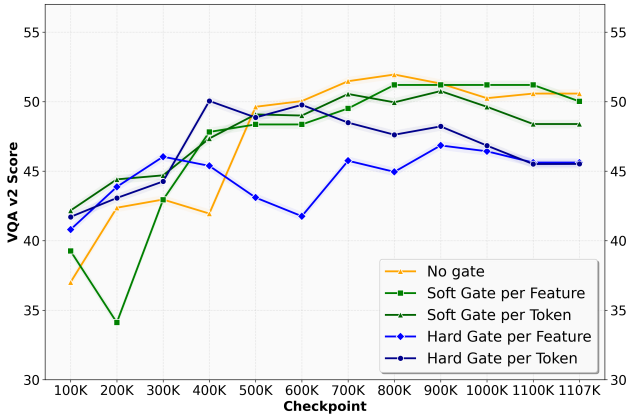
(b) BLiMP Supplement scores.



(c) EWoK scores.



(d) Winoground scores.



(e) VQA scores.

Figure 5.1: The performance of my base model with different gate variants, *no gate*, *soft gate per feature*, *soft gate per token*, *hard gate per feature*, *hard gate per token*. The graphs illustrate the checkpoints' performance saved every 100,000 steps on the BLiMP, BLiMP Supplement, EWoK, Winoground and VQA benchmarks. The shading around the graph lines represents the standard error in the evaluation.

other models, with the *soft gate per token* variant scoring between 0.5%-1.5% less in all checkpoints. However, given that all models have a similar curve, I cannot attribute this difference with complete certainty to the gating module, as it could also be due to a run variation. Overall, all models score between 74.69% and 75.53% on BLiMP by the end of the 10 training epochs, and I conclude that my dynamic gating modules do not have a significant effect on the model's performance on BLiMP

beyond the 200,000 steps checkpoint.

BLiMP Supplement. In the case of BLiMP Supplement, subfigure 5.1b shows that the dynamic gate modules introduce more performance instability. As I further discuss in Section 5.5, some of this instability is attributed to the alternation between text-only and image-caption epochs during training. However, the *no gate* model is able to stabilise after 600,000 steps regardless of the training regime, while the dynamic gating models' performance oscillates until the end of training. After the 10 training epochs, the *soft gate* models obtain similar scores to the *no gate* version on BLiMP Supplement, while *hard gate* variants achieve a lower score by 1%-1.75%. Combining these results with the analysis that I perform in Section 5.5, I conclude that the gating modules introduce more performance instability when paired with my training regime in the case of BLiMP Supplement, yet the *soft gate* models are able to achieve a similar score to the *no gate* variant by the end of training.

EWoK. As shown in subfigure 5.1c, the EWoK benchmark is not suited to analyse the impact of my dynamic gating modules. There is very little variation in the EWoK scores across both models and checkpoints. This is a problem that extends beyond my dynamic gating analysis, as I encounter this pattern in the evaluation of all my models. I discuss further in Chapter 6 that the training data do not well support the EWoK benchmark, as a significant proportion of the concepts evaluated in the benchmark are present too few times in the training datasets. I thus conclude that, as expected, there is no impact of the dynamic gating modules on the EWoK scores given the BabyLM Challenge training data.

Winoground. The results in 5.1d show a modest yet meaningful benefit of the dynamic gating modules for the Winoground benchmark. After 500,000 steps of training, the *soft gate per feature*, *soft gate per token* and *hard gate per token* models constantly outperform the *no gate* version. These models also outperform the Flamingo baseline on Winoground for multiple checkpoints, including the final one. The *hard gate per feature* model exhibits more unstable performance, suggesting that the gate of this model creates noisier, possibly mismatching feature representations. I hypothesise that the gating mechanism in the *soft gate per feature*, *soft gate per token* and *hard gate per token* models produces slightly cleaner, more discriminative joint representations between images and text than the *no gate* model. That, in turn, yields a small but consistent improvement on Winoground.

VQA. Subfigure 5.1e illustrates the scores of my base model and its dynamic gate variants on VQA. The aim of adding a dynamic gating module was to enable the model to make more fine-grained decisions when predicting the next token. Therefore, I hypothesised that this mechanism would positively impact the performance score of my base model on VQA, by enabling it to discriminate better between the correct answer and ungrammatical and implausible distractor answers. However, I observe limited performance benefits of the gating modules on VQA, given my training data and training regime. By the 500,000th training steps the gate variants of the base model achieve higher VQA scores than the *no gate* version at most checkpoints. However, after this step, the *hard gate* models become more unstable in their performance, achieving a 5% lower score than the *no gate* base model

by the end of training. The *soft gate* versions follow a more similar curve to the *no gate* model, with the *soft gate per feature* variant achieving a similar score at the end of the 10th epoch (50.58% and 50.02% respectively), and the *soft gate per token* achieving a lower score by 2%.

I deduce that the *hard* and *per token* gate variants negatively impact the performance of my model on the VQA benchmark, given my training data and training regime, and suspect a connection with the training data. More specifically, I notice and analyse in section 5.5 that the model's performance on VQA decreases after image-caption epochs. This, to a certain extent, is expected, as the image-caption dataset does not contain any turn-taking constructions and many fewer questions than the text-only dataset. This contrasts with the format of the questions in the VQA dataset. Therefore, one theory is as follows: the cross-attention module in the models having the *hard* and *per token* gates may have learned during training to allow for stronger image signals in its fused representation, as the dynamic gate also adds a strong text signal after. However, these models have learned to do so on image-caption datasets that do not include questions or turn-taking. Therefore, when encountering a question paired with an image, the gate could overemphasise the visual features past the optimal values for VQA, treating the question tokens as if they were descriptive captions, and thereby suppress the linguistic reasoning required to interpret the question. As a result, the model focuses on irrelevant image content instead of parsing the question, leading to a drop in VQA accuracy.

Conclusions. I conclude that, given my current training data and training regime, the dynamic gating modules maintain the performance of my base model without gating on BLiMP and BLiMP Supplement, show mixed results on VQA, and bring modest benefits for Winoground. However, I do not discard the use of dynamic gating for multimodal BabyLMs. As I discuss in Section 5.6, there is a correlation between gate selections and parts-of-speech, which suggests that gating may emulate aspects of human selective attention. Moreover, I argue in Chapter 6 that training a multimodal BabyLM on data that is suited for VQA could lead to better results on this benchmark in general. This case would require a re-implementation of the evaluation for the different flavours of gating I propose in this work.

5.2.2 Feature Representation

To address the limited representational capacity of using only a global *CLS* image embedding, I investigate whether feature modulation can enhance the textual and visual representation in my framework, and implicitly the performance of my base model. Figure 5.2 summarises the absolute performance scores and difference compared to the base model for Feature-wise Linear Modulation (FiLM) (Perez et al., 2018) and Dynamic Intra-modulation (DyIntra) (Gao et al., 2019) applied at different points in the architecture (text stream, image stream, and cross-attention output), as well as channel attention on the image features. As shown, across the seven variants, no single feature representation technique uniformly improves all five benchmarks.

Result 1: Several modulation variants demonstrate modest improvements on specific benchmarks. FiLM applied to textual representation and cross-attention, along with channel attention

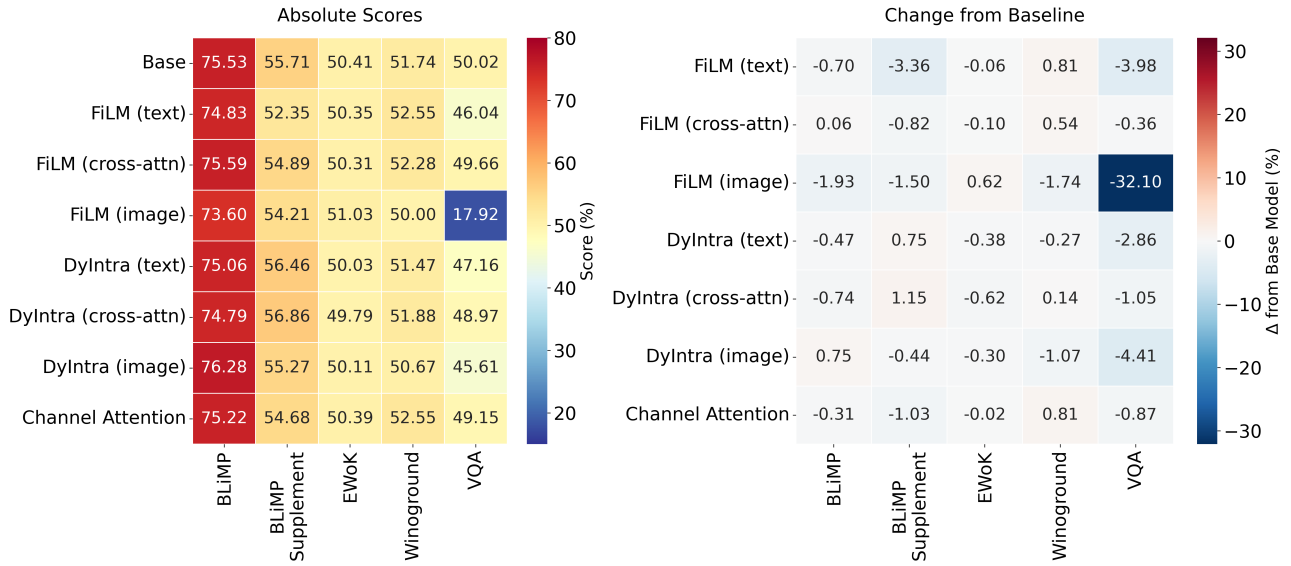


Figure 5.2: The absolute performance scores and difference compared to the base model for Feature-wise Linear Modulation (FiLM) (Perez et al., 2018) and Dynamic Intra-modulation (DyIntra) (Gao et al., 2019) applied at different points in the architecture (text stream, image stream, and cross-attention output), as well as channel attention on the image features.

applied to the image, show the highest gains on Winoground (+0.81%, +0.54%, and +0.81% respectively) by potentially creating more separable joint representations. FiLM applied to the image has the highest positive impact on EWoK, being the only version of my models that is able to surpass 51% on this benchmark. However, this comes at the severe cost of performance on the other benchmarks, particularly VQA. The 32.10% score drop suggests that modulating the already compressed *CLS* token corrupts the limited visual information it contains, which makes the model unable to ground answers in visual input.

Result 2: Except for FiLM on the image, the differences on BLiMP are not significant across techniques. As shown on the right side of figure 5.1, the BLiMP score differences between the base model and the modulation and channel attention variants lie between -0.75% and 0.75% (with the exception of FiLM on the image), which may be due to run variations. For BLiMP Supplement, FiLM on text shows a clear decrease in performance (-3.36%), while other variants show mixed results.

Result 3: All modulation variants, as well as the channel attention method, have lower performance on VQA than the base model. Although with varied impact, all techniques decrease the performance of my base model on VQA. Nevertheless, the cross-attention modulation and channel attention seem to preserve most of the linguistic and visual signals needed for VQA, while also bringing slight improvements to Wionground.

Overall, modulation and channel attention achieve mixed results over the five benchmarks, underscoring that the *CLS* image embedding represents a performance bottleneck. The results collectively demonstrate that feature modulation and channel attention techniques designed for rich representations show limited and task-specific benefits when applied to severely compressed representations. While certain combinations can enhance performance on specific benchmarks, they cannot overcome the information bottleneck caused by using only a global *CLS* image embedding.

5.3 Performance of Auxiliary Objective Functions

Total Loss Function	Aux. Func. Weight	**BLiMP**	**BLiMP Supplement**	**EWoK**	**Winoground**	**VQA**
NTP		**75.53 ± 0.16**	**55.71 ± 0.57**	50.41 ± 0.57	**51.74 ± 1.83**	**50.02 ± 0.31**
NTP + CLIP	1.0	71.96 ± 0.16	55.12 ± 0.55	**50.83 ± 0.57**	51.21 ± 1.83	47.72 ± 0.31
NTP + LCG	0.3	70.33 ± 0.17	52.88 ± 0.55	49.57 ± 0.57	51.21 ± 1.83	41.82 ± 0.31

Table 5.2: My base model's performance with **next token prediction (NTP)** as the main loss function and **contrastive learning (CLIP)** and **LexiContrastive Grounding (LCG)** as auxiliary losses. The vanilla training regime uses a batch size of 64. The auxiliary function variants use a batch size of 128.

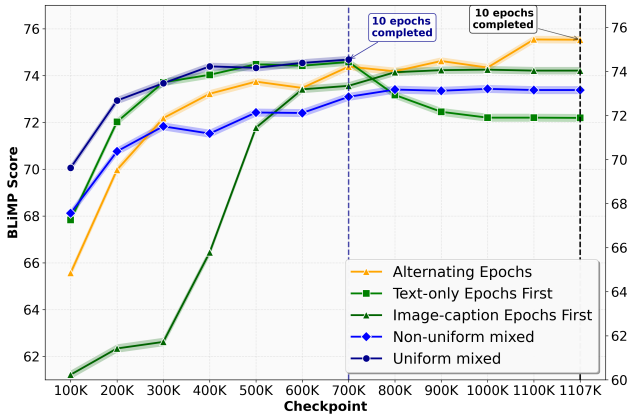
In table 5.2, I summarise the performance scores of my base model after 10 epochs of training, using next token prediction (NTP) as the main objective function and contrastive learning (CLIP) (Radford et al., 2021) or LexiContrastive Grounding (LCG) (Zhuang et al., 2024) as the auxiliary loss function. I use a batch of 64 for the former training regime and a batch size of 128 for the latter, as a higher batch size is recommended for contrastive learning (Chen et al., 2020). I select λ, the weight of the auxiliary function, through trial and error such that the values of the main and auxiliary objective functions are of similar magnitudes.

As shown, a pure next token prediction objective function achieves the best scores for my base model overall. There are multiple potential causes for these results. First, the BLiMP benchmarks rely solely on linguistic information, therefore any auxiliary objective that competes with NTP can dilute the model's focus on linguistic signals. This is reflected in the BLiMP score differences of 3.57% and 5.2% with the CLIP and LCG auxiliary functions, respectively. Second, the CLIP objective was designed for a larger batch size than I could use with my computational budget and more data than the available samples in the image-caption dataset, which may have led to a limited impact. Third, the global image embeddings provide limited visual information, which seems to be insufficient to enable CLIP to make fine-grained visual-linguistic alignments and LCG to achieve word-level grounding. Fourth, the alternation between text-only and image-caption epochs may cause training instability, since the auxiliary functions are only used during the image-caption epochs.

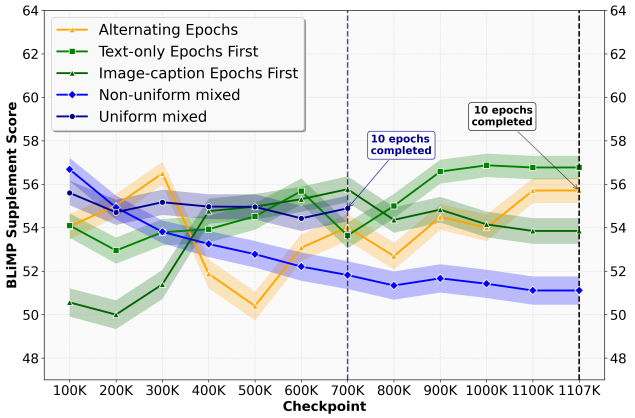
Therefore, with the current fixed constraints of using only 10 epochs of training and the limited global image embeddings, there is no evident benefit of using contrastive learning auxiliary objectives. CLIP and LCG decrease my base model's performance on BLiMP and VQA (and BLiMP Supplement for LCG), and maintain it on EWoK and Winoground.

From a cognitive perspective, these negative results may align better with theories of human language acquisition. Children do not learn language through explicit contrastive mechanisms where they simultaneously process what words do and do not mean across hundreds of examples. Words are learned in rich, multimodal contexts where meaning emerges from use rather than from explicit positive or negative examples. These results support my focus on architectural innovations, such as dynamic gating, which better capture the selective and adaptive nature of human cognitive processing during language learning.

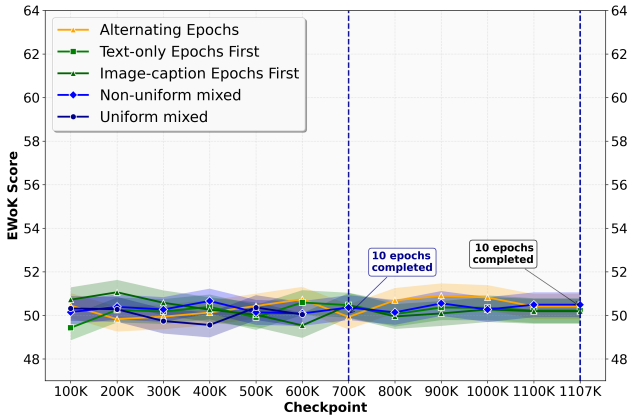
5.4 Effect of Data Curriculum



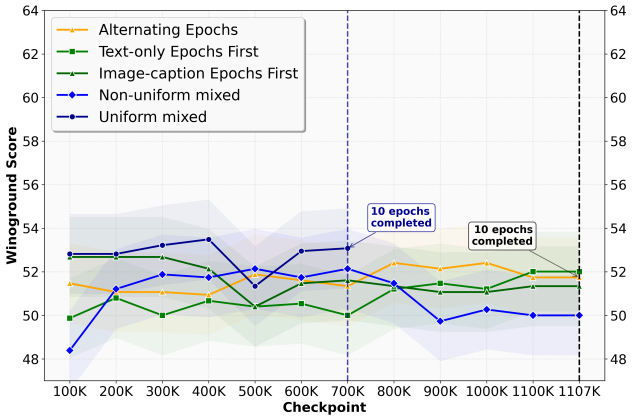
(a) BLiMP scores.



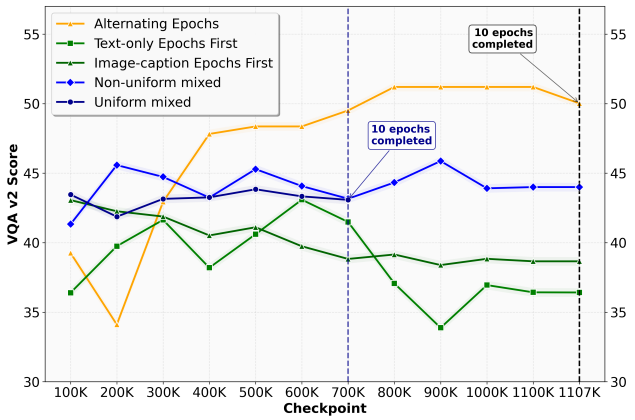
(b) BLiMP Supplement scores.



(c) EWoK scores.



(d) Winoground scores.



(e) VQA scores.

Figure 5.3: The performance of my base model on BLiMP, BLiMP Supplement, EWoK, Winoground and VQA for different data curriculum strategies. The *uniform mixed* strategy follows a different definition than the others, where the number of steps in an epoch equals the number of text-only samples. This results in ∼700K training steps for 10 epochs, which are marked in the graphs by the blue dashed line. The end of 10 epochs for the other data curriculum strategies is marked by the black dashed line at step ∼1107K.

Note: In my analysis, I refer to a complete pass over the text-only dataset and image-caption dataset (a total of 100M words) as *epoch*, a pass over the text-only dataset (a total of 50M words) as *text-only epoch* and a pass over the image-caption dataset (a total of 50M words) as *image-caption epoch*.

Therefore, *10 epochs* consist of *10 text-only epochs* and *10 image-caption epochs*.

Figure 5.3 visualises the scores of my base model for the data curriculum strategies I define in this work. I refer to them as (1) *alternating epochs*: alternating between 10 text-only epochs and 10 image-caption epochs; (2) *text-only epochs first*: training the model on 10 text-only epochs first, then on 10 image-caption epochs; (3) *image-caption epochs first*: training the model on 10 image-caption epochs first, then on 10 text-only epochs; (4) *non-uniform mixed*: training the model on 10 epochs where the training batches contain both text-only and image-caption samples distributed non-uniformly, equalling 10 text-only epochs and 10 image-caption epochs; (5) *uniform mixed*: training the model on 10 epochs where the training batches contain both text-only and image-caption samples distributed uniformly, equalling 10 text-only epochs and 20 image-caption epochs. A detailed explanation of these strategies is available in section 3.7.

As shown, **the results over the five benchmarks validate the choice of alternating between text-only and image-caption epochs in my framework**.

BLiMP. Subfigure 5.3a illustrates the BLiMP scores for the different data curriculum strategies. The *alternating epochs*, *non-uniform mixed* and *uniform mixed* strategies follow a consistent pattern, where the score improves over checkpoints. In contrast, the *text-only epochs first* and *image-caption epochs first* show drastic changes when switching between epoch types. The pattern of the two suggests that (1) the text-only dataset supports the BLiMP benchmark far more than the image-caption one, and (2) these strategies result in catastrophic forgetting for the model by the end of training. The consistent improvement over checkpoints and final score of alternating between epoch types prove that this strategy is the best for my base model on the BLiMP benchmark.

BLiMP Supplement. For BLiMP Supplement, the optimal data curriculum strategy is less clear than it is for BLiMP. The model's performance oscillates when alternating between epoch types, as detailed in section 5.5. Comparing the *text-only epochs first* and *image-caption epochs first* strategies shows that the image-caption dataset better supports the model on BLiMP Supplement than the text-only dataset, which aligns with further results in section 5.5. Interestingly, the model's performance score consistently decreases over checkpoints when the model is trained using the *non-uniform mixed* strategy. A possible explanation for this result is that since there are more text-only samples in a batch than image-caption samples, the gradient updates are dominated by the text-only data, reducing the effect of the image-caption samples. The performance pattern for the *uniform mixed* strategy remains consistent, however, resulting in a lower final score than the *alternating epochs* and *text-only epochs first* variants.

EWoK. Similar to the other analyses in this work, changing the data curriculum strategy has no visible effect on the EWoK benchmark, underscoring that the training data might not be well-suited for this benchmark.

Winoground. As shown in subfigure 5.3d, training my base model using the *uniform mixed* strategy results in a higher Winoground score, with several checkpoints achieving over 53% on this benchmark. However, a significant factor contributing to this result is the amount of image-caption training

data, which is double for this strategy than for the others. Comparing the *text-only epochs first* and *image-caption epochs first* strategies, it can be seen that the model performs better on Winoground when consistently trained on the image-caption dataset. Using the *non-uniform mixed* strategy results in a more unstable performance and a lower final score, possibly due to the dominance of text-only samples in the training batches. There is a slight increase in performance across checkpoints using the *alternating epochs* strategy, with the model achieving a competitive Winoground score by the end of training.

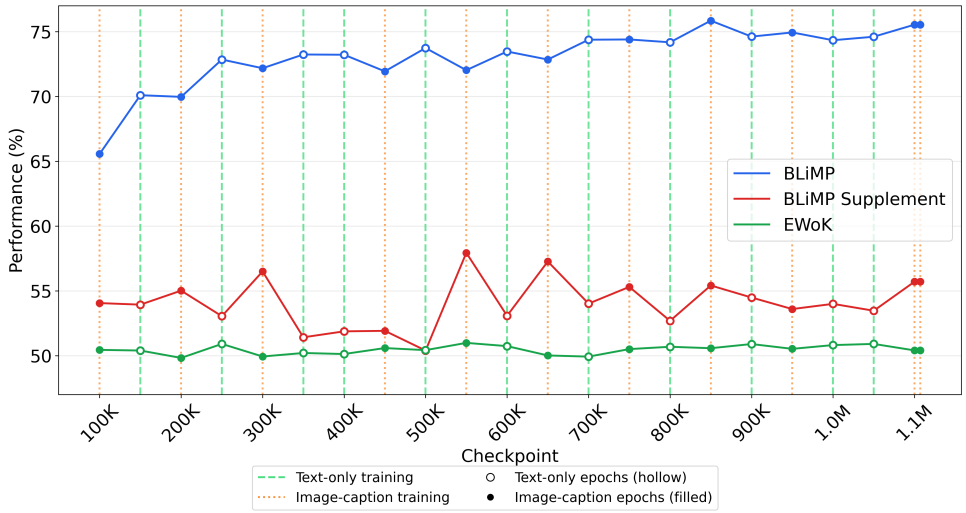
VQA. As shown in subfigure 5.3e, the alternating between text-only and image-caption epochs strategy achieves the best performance on VQA. There is a significant performance gap between the model trained using *alternating epochs* compared to the *mixed* strategies (over 5%), as well as the *text-only epochs first* and *image-caption epochs first* strategies (over 10%). The *alternating epochs* strategy shows an almost consistent increase over checkpoints, whereas the model's performance in the *mixed* variants remains flat, and decreases for the *text-only epochs first* and *image-caption epochs first* strategies. The results of the coarse-grained strategies are likely due to the training data. The image-caption dataset supports the visual reasoning component of VQA, while the text-only dataset supports the question format by containing turn-taking constructions and a significantly larger number of questions than the other dataset. Training the model consistently on only one epoch type deprives it of one of these complementary components. Training by alternating between epoch types appears to strike a balance and avoid catastrophic forgetting. The results of the *uniform mixed* strategy are slightly surprising given that the Flamingo and GIT baselines achieve higher VQA scores using this approach, however, the difference could stem from using a lower learning rate and fewer training epochs.

In conclusion, the model's performance across all benchmarks indicates that the alternating-between-epochs type is the optimal data curriculum strategy for my framework in the context of the BabyLM Challenge. These results motivate further analysis of the model's training dynamic using this approach, which I investigate in the following section.

5.5 Training Dynamics

In figure 5.4, I visualise the performance of my base model, evaluated every 50,000 steps on BLiMP, BLiMP Supplement, EWoK, Winoground and VQA, when alternating between text-only and image-caption epochs. The brown dotted lines indicate that the checkpoint was saved during a text-only epoch, while the green dashed lines indicate that the checkpoint was saved during an image-caption epoch. For BLIMP Supplement and VQA, an interesting pattern emerges: the performance scores significantly oscillate based on the type of data on which the model was last trained.

The data that the model was last trained on can be regarded as a fine-tuning step. Thus, I make the following observations:



(a) BLiMP, BLiMP Supplement and EWoK scores.

(b) Winoground and VQA scores.

Figure 5.4: The performance of my base model every 50,000 steps on the BLiMP, BLiMP Supplement, EWoK, Winoground and VQA benchmarks. The brown doted lines indicate that the checkpoint was saved during a text-only epoch, while the green dashed lines indicate that the checkpoint was saved during an image-caption epoch.

Observation 1: The base model achieves better performance on BLiMP Supplement during image-caption epochs. As shown in subfigure 5.4a, my base model obtains higher scores on BLiMP Supplement, a text-only benchmark evaluating grammar, at checkpoints saved during image-caption epochs compared to text-only epochs. I, therefore, investigate the breakdown of the BLiMP Supplement scores and notice that the score difference for different epoch types stems from two subtasks, *subject-auxiliary inversion* and *turn-taking*. For these subtasks, the performance of my base model fluctuates by even $\sim$10% between checkpoints. By analysing the individual log-likelihood scores for each test sample for two different checkpoints, I determine that specific patterns in the image-caption dataset coincidentally facilitate a higher probability for many of the sentences labelled as correct in the *subject-auxiliary inversion* subtask. For the *turn-taking* subtask, although there is no noticeable pattern in the training dataset, I observe that the model selects the correct sentences with little confidence. A complete description of this investigation and findings is available in Appendix D.

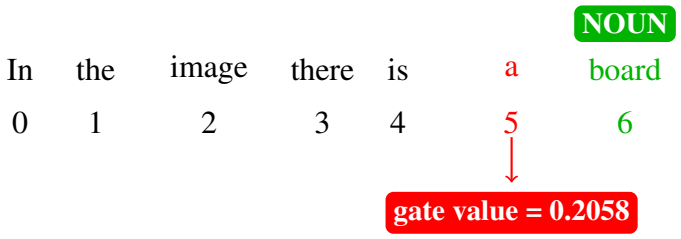
Observation 2: The base model achieves better performance on VQA during text-only epochs. In figure 5.4b, it can be noticed the score of my base model on VQA oscillates by 5% to 10% between text-only epochs and image-caption epochs. I theorise that the cause of these variations is the difference in textual data between the two types of epochs. There are no turn-taking constructions in the image-caption datasets, and the number of questions (25,300 question marks) is significantly lower than in the text-only datasets (1,083,559 question marks). However, both are present in the format of the VQA text data. Therefore, I conclude that the image-caption datasets support the VQA task less due to differences in the text format. I argue that for a high score on VQA during image-caption epochs, the image-caption datasets should contain samples similar to the task.

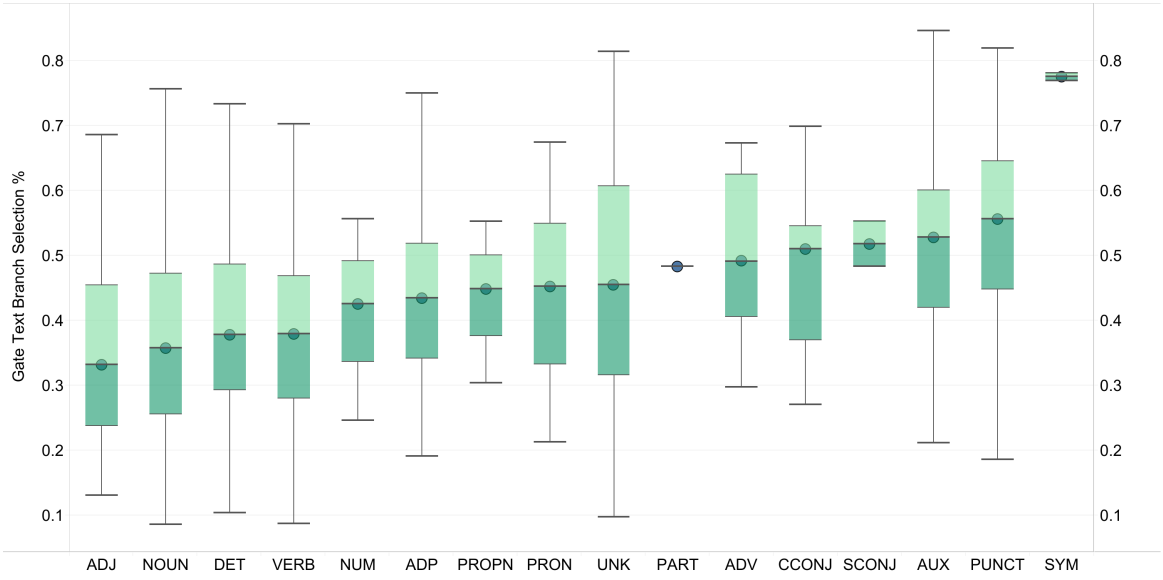
Observation 3: The alternation between text-only and image-caption epochs has little to no effect on the BLiMP, EWoK and Winoground benchmarks for the base model. As shown in figure 5.4, there is little oscillation between text-only and image-caption epochs on the BLiMP benchmark, suggesting that the text-only dataset supports the model better for this task, but the score generally increases. There are no noticeable patterns for EWoK or Winoground.

Note: The reason the scores in all benchmarks stabilise after checkpoint 800,000 is because of the small learning rate (5e-5) combined with the learning rate schedule (cosine annealing) I chose for training. After checkpoint 800,000, the learning rate gradually decreases from 1e-5 to 0, which has little effect on the gradients.

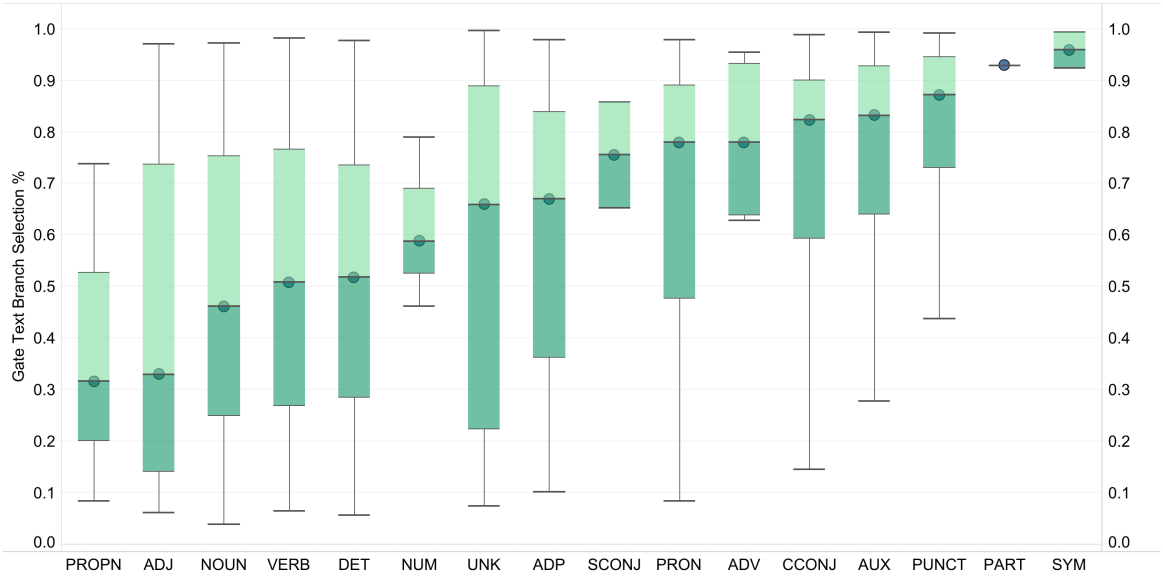
5.6 Interpretability and Correlation to Parts-of-Speech

Figure 5.5 illustrates the model's gate value for next token prediction for the *soft gate per feature* and *soft gate per token* model variants, aggregated per part-of-speech (PoS). The models are evaluated on held-out test sentences from the Localized Narratives dataset, accounting for 1,034 tokens. In the case of the *soft gate per feature* variant, the gate value plotted is the mean over the gate values per feature. A lower score means that the model attended less to the pure linguistic signals and more to the fused image-text representation when predicting the next token. For example, for the sentence "In the image there is a board", there is the tuple $(word = board, gate = 20.58\%, PoS = noun)$, where the gate value at position 5, corresponding to "a", determines the mixture of textual features and fused (cross-attention) features when predicting the token at position 6, corresponding to "board":





(a) Soft gate per feature.



(b) Soft gate per token.

Figure 5.5: The gate value for next token prediction for the *soft gate per feature* and *soft gate per token* gate variants, evaluated on sentences from the Localized Narratives dataset. A lower score means that the gate attended less to the pure text hidden representation and more to the fused image-text (cross-attention) hidden representation. The blue dots represent the median over the gate values corresponding to each part-of-speech. The green boxes show the interquartile range, spanning from the 25th to 75th percentile of gate values. The whiskers stretch to the minimum and maximum values.

As shown in figure 5.5, there is a cognitively-motivated correlation between gate selection and part-of-speech present in both *soft gate* variants. For the parts-of-speech which are open-class and generally more grounded (adjective, noun, proper noun, verb), both models attend more to the image signals (left side of the plots), while for function words (conjunction, punctuation, symbols, auxiliary verbs, particles) the models attend more to the pure text (right side of the plots). Furthermore, the models show increased visual grounding for numerals, determiners and adpositions, suggesting they leverage visual information for counting and quantity (“two”, “three”), uniqueness (“a” vs “the”), spatial reference (“this” vs “that”) and spatial relationships (“on”, “in”, “around”).

Comparing the patterns of the two soft gate variants, the fine-grained mechanism of *soft gate per feature* model enables it to make more confident and consistent gating decisions. This can be seen in the difference between the interquartile and min-max ranges of the two gates. Per part-of-speech, the interquartile range for *soft gate per feature* values spans at most 25% (excluding UNK), while for the *soft gate per token* it spans up to 60% in the case of adjectives. These results suggest that the *per feature* model learns more stable and generalisable gating policies for when to use visual features within each parts-of-speech class. This is consistent with the models' evaluation on the BabyLM Challenge benchmarks, where the *soft gate per feature* variant achieves higher performance than the *soft gate per token* one, as discussed in section 5.2.1.

I confirm the correlation between gate selection and parts-of-speech by running the Kruskal-Wallis statistical test (McKight and Najab, 2010) for both *soft gate* variants. For the *soft gate per feature*, I obtain $(H = 154.91, p < 0.001)$ and for the *soft gate per token*, I obtain $(H = 164.43, p < 0.001)$.

For the *hard gate* variants, their relationship to parts-of-speech is less interpretable. The *hard gate per feature* model's scores equal the proportion of features for which the gate selected text over visually-enriched representations. Without knowing what individual features represent, these scores provide limited insight. The plot for the *hard gate per feature* gate selection is available in Appendix F. The *hard gate per token* model presents a different challenge. It learns to almost exclusively select the fused text-image representation (the gate value per token becomes 0), therefore bypassing the gating mechanism and becoming a standard multimodal model. This suggests that the discrete decisions of the *hard gate per token* model may be too restrictive for learning nuanced modality selection policies.

Similarly to parts-of-speech, I also investigate whether there is a correlation between *soft gate* values and concreteness, familiarity and imageability scores, as well as age of acquisition. For this, I use the MRC Psycholinguistic Database (Coltheart, 1981) and directly map the next word to be predicted with its corresponding score in the database. I then use the Spearman's rank correlation test to asses the relationship between gate values and each psycholinguistic measure. I find a statistically significant negative correlation between concreteness and gate values as well as between imageability and gate values for both *soft gate* variants.

Table 5.3 summarises the Spearman's test results, while table 5.4 aggregates the *soft gate per feature* and *soft gate per token* values per concreteness and imageability categories. In table 5.4, I define bins for concreteness and imageability based on their distribution in the MRC database to create psychologically meaningful categories. Specifically, I use cutpoints at mean ± 1 SD to separate words into four groups: Very Abstract/Low ($< \mu - \sigma$), Abstract/Low ($\mu - \sigma$ to μ), Concrete/High (μ to $\mu + \sigma$), and Very Concrete/Very High ($> \mu + \sigma$), where μ and σ are the mean and standard deviation reported in the MRC database documentation.

As shown, more concrete and imageable words receive more visual grounding (lower gate values) in both *soft gate* variants. However, the correlation is weak ($|\rho| < 0.2$), and the pattern is non-monotonic i.e., moderately abstract/concrete words show higher gate rates than the very abstract/concrete ones, suggesting that other factors, such as part-of-speech, are more important in gating decisions.

Measure	Soft Gate per Feature	Soft Gate per Token
Concreteness	$\rho = -0.139, p < 0.001$	$\rho = -0.156, p < 0.001$
Imageability	$\rho = -0.153, p < 0.001$	$\rho = -0.151, p < 0.001$

All correlations based on n=701 (concreteness) and n=715 (imageability) words.

Table 5.3: Spearman correlations between gate values and psycholinguistic properties for the *soft gate* variants.

Measure	Category	Soft Gate per Feature		Soft Gate per Token	
		Mean (SD)	# words	Mean (SD)	# words
Concreteness	Very Abstract (<318)	0.427 (0.141)	420	0.586 (0.273)	420
	Abstract (318-438)	0.471 (0.136)	82	0.674 (0.248)	82
	Concrete (438-558)	0.391 (0.155)	80	0.529 (0.292)	80
	Very Concrete (>558)	0.343 (0.139)	119	0.428 (0.267)	119
Imageability	Very Low (<342)	0.427 (0.143)	401	0.588 (0.273)	401
	Low (342-450)	0.481 (0.130)	96	0.686 (0.238)	96
	High (450-558)	0.378 (0.126)	78	0.509 (0.276)	78
	Very High (>558)	0.351 (0.155)	140	0.441 (0.281)	140

Categories defined as $\mu \pm 1\sigma$ based on MRC database references. SD = standard deviation.

Table 5.4: Mean gate values by psycholinguistic categories for the *soft gate* variants.

In conclusion, there is a link between the *soft gate* mechanisms proposed in my framework and human cognition. The models learn to distinguish between content words that tend to require visual grounding (nouns, verbs, adjectives) and function words that tend to require mainly linguistic signals (conjunctions, auxiliaries, particles) without explicit supervision. The *soft gate per feature* values exhibit more consistent patterns, indicating more stable gating policies, which in turn translate to higher performance on the BabyLM Challenge evaluation benchmarks.

While there is a statistically significant correlation between gating decisions and concreteness and imageability scores, this correlation is weak ($|\rho| < 0.2$). These results indicate that grammatical category is the primary factor for modality selection and suggest that the model prioritises syntactic over semantic cues when determining visual grounding. This finding raises questions about whether incorporating stronger semantic cues would result in stronger correlations to psycholinguistic metrics and, implicitly, better multimodal performance, possibly by using patch-token image representations instead of a global *CLS* token.

Chapter 6

Discussion

In this chapter, I summarise the key findings from my results and discuss their implication for multimodal language learning under the BabyLM Challenge constraints.

6.1 Key Findings

6.1.1 Dynamic Gating and Cognitive Plausibility

One of the most significant findings in this work is that dynamic gating mechanisms can learn cognitively-plausible patterns of multimodal fusion without explicit supervision, demonstrating a strong correlation between gate selections and parts-of-speech (Section 5.6).

However, these mechanisms showed modest and task-specific improvements on the BabyLM Challenge benchmarks. This gap between cognitive plausibility and performance improvement raises important questions:

1. To what extent are the BabyLM Challenge evaluation benchmarks able to capture the benefits of cognitively-inspired architectures?
2. Other than the ones explored here, which cognitive mechanisms are necessary to achieve significant performance improvements?
3. Can cognitively-inspired architectures bridge the gap when the underlying mechanisms of machine learning (e.g., gradient descent) and human learning remain fundamentally different?

6.1.2 Global Image Embeddings as an Information Bottleneck

The limited impact of the feature enhancement techniques I explore in this work, FiLM (Perez et al., 2018), DyIntra (Gao et al., 2019) and channel attention, underscores a significant constraint: global *CLS* tokens provide insufficient visual information for fine-grained multimodal learning. This finding has implications for the BabyLM Challenge design, as this constraint may restrict models to superficial multimodal fusion, and therefore, may unreasonably limit participants in the Challenge.

6.1.3 Auxiliary Objectives and Human Language Learning

Another important finding in this work is the negative results of using contrastive learning auxiliary objectives, CLIP (Radford et al., 2019) and LCG (Zhuang et al., 2024). These methods were unable to improve the base model's performance on the benchmarks under the BabyLM Challenge constraints, as they require large batch sizes and rich visual representations to achieve fine-grained alignments. From a cognitive perspective, these results may be more aligned with human language acquisition. Children do not learn language through explicit contrastive mechanisms where they simultaneously compare a large number of positive and negative examples. Instead, they learn through rich visio-linguistic input where meaning emerges from use. This suggests that architectural innovations such as dynamic gating, which selectively incorporates attention cues, may be more cognitively appropriate than auxiliary objective functions that require large parallel comparisons.

6.2 Training Data and Evaluation Benchmarks

Missing modality problem. As results in 5.5 suggest, the split of the training data into text-only and image-caption datasets introduces complexity and instability during training. While I attempt to mitigate this in my work by alternating epochs, this approach still yields performance oscillations. The BabyLM Challenge 2024 baselines addressed this problem by pairing text-only and image-caption in the same batch. However, this required training the models on twice as many image-caption samples, which conflicts with the 10-epoch limit introduced in 2025. Moreover, to the best of my knowledge, there is no cognitive justification for this split.

Benchmark suitability in connection to training data. Two of the BabyLM Challenge benchmarks I used in this work showed limitations in evaluating my multimodal models, which potentially stem from a mismatch with the training data.

EWoK demonstrates no sensitivity to changes in architecture or training strategy, with performance remaining around 50% regardless of the experiment conditions. This indicates a mismatch between the concepts tested in EWoK and those present in the training dataset. I therefore investigate the frequency of concepts tested by EWoK in the BabyLM Challenge training data, as previous research suggests that language models rely on frequency more than children do in word acquisition (Chang and Bergen, 2022). EWoK is constructed such that the model is presented with two target sentences differing by one or a few words, denoted as concepts. I perform a Regular Expression match for the concepts tested in EWoK over the BabyLM Challenge training data. I find that in 37.69% of the EWoK test examples, at least one concept of the two targets appears fewer than 100 times in the training data, with 13% of test examples having both concepts appearing 0 times. Therefore, I conclude that the training dataset does not properly support EWoK evaluation.

As discussed in Section 5.5, the score difference between epoch types for VQA suggests that this benchmark depends significantly on the presence of question-answer and turn-taking formats in the training dataset rather than on visual understanding capabilities. These VQA performance patterns

align with observations by Laurençon et al. (2024a), who note that vision-language models typically only learn visual question answering during fine-tuning stages, not during pre-training, unless they are explicitly exposed to data following the VQA format. This is particularly problematic under the BabyLM constraints where no fine-tuning stage exists, forcing models to acquire question-answering capabilities just from pre-training data that lacks examples similar to the VQA task.

Chapter 7

Summary and Conclusions

In this work, I explored several cognitively-inspired approaches to multimodal language learning under the constraints of the BabyLM Challenge. By developing a framework that incorporates dynamic gating, feature enhancement techniques and contrastive learning auxiliary objectives, I investigated whether models can learn language more like humans do.

My key contributions include:

- **Dynamic gating as selective attention**: The token-wise dynamic gating mechanisms I introduced in this work supported the learning of cognitively-plausible data fusion patterns without supervision. Models learned to attend more to visual cues for content words (nouns, adjectives, verbs) while prioritising linguistic signals for function words (auxiliary verbs, particles, conjunctions). While performance improvements on the BabyLM Challenge benchmarks were modest, the interpretable gating mechanism seems to mirror aspects of human selective attention;
- **Identifying limitations of the BabyLMs Challenge Vision track setup**: My experiments revealed that using a single global image embedding becomes a severe limitation for multimodal learning. Feature enhancement techniques were insufficient to overcome this bottleneck, highlighting that the setup of the BabyLM Challenge Vision track may not be optimal. Moreover, some of my findings raised questions about the suitability of the training data and evaluation benchmarks;
- **Evaluating training strategies**: Based on the data curriculum strategies I defined in this work, I found that alternating between text-only and image-caption epochs achieves the best results across benchmarks, at the expense of training stability. The negative results from using contrastive learning auxiliary objectives suggest that learning via large-scale parallel comparisons may be less cognitively aligned.

Despite current limitations, the framework I introduce in this work represents a crucial step toward language models that learn not only what humans know but also how humans learn: efficiently, incrementally and grounded in visual experience.

7.1 Limitations and Future Work

As discussed in various sections of this work, the global *CLS* image embeddings proved to be a significant limitation in assessing the capabilities of my proposed method, with feature enhancement techniques being insufficient in improving the image representation (Section 5.2.2). Moreover, the training data was a significant factor influencing the models' results on the BabyLM benchmarks, which may not fully reflect the architectural features I introduced in this work.

As a general observation, training vision-language BabyLMs differs from training state-of-the-art large vision-language models, which rely on large pre-trained components. Moreover, VLMs can undergo multiple training stages where components are selectively frozen or unfrozen, higher-quality data is gradually introduced and the image resolution is progressively increased (Laurençon et al., 2024a). With limited data and a maximum of just 10 training epochs under the BabyLM Challenge constraints, implementing multi-stage training strategies becomes significantly more difficult.

Based on the results and findings in this work, for the BabyLM Challenge Vision track, I make the following recommendations and observations for future work:

- The training dataset should be varied, with high-quality text that covers a range of English constructions. In particular, the dataset should cover constructions present in the evaluation benchmarks (for example, images paired with question-answers for VQA), as well as ensuring that the training data covers (with a certain threshold) the concepts present in the test data (e.g., EWoK);
- For training stability and improved language acquisition, it may be more beneficial to train the model on a completely multimodal dataset, which is an interesting lead to investigate in future work;
- Given the limitations that the global *CLS* token introduced in this work, future work should use patch-token representations for the image input, which is the aim of future iterations of this framework. This would enable richer multimodal learning and potentially benefit from word-level contrastive learning;
- While the BabyLM Challenge evaluates overall language acquisition, it would be interesting to develop benchmarks that specifically reward cognitively-plausible mechanisms, i.e., evaluating not only what the model produces, but also the cognitive principles guiding its responses.

Bibliography

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.

Badr AlKhamissi, Yingtian Tang, Abdülkadir Gökce, Johannes Mehrer, and Martin Schrimpf. Dreaming out loud: A self-synthesis approach for training vision-language models with developmentally plausible data. In Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox, editors, *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 244–251, Miami, FL, USA, November 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.conll-babylm.22/.

Andrew J Anderson, Douwe Kiela, Stephen Clark, and Massimo Poesio. Visually grounded and textual semantic models differentially decode brain activity associated with concrete and abstract nouns. *Transactions of the Association for Computational Linguistics*, 5:17–30, 2017.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. URL https://arxiv.org/abs/2308.12966.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2): 423–443, 2018.

Tyler A Chang and Benjamin K Bergen. Word acquisition in neural language models. *Transactions of the Association for Computational Linguistics*, 10:1–16, 2022.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, et al. Babylm turns 3: Call for papers for the 2025 babylm workshop. *arXiv preprint arXiv:2502.10645*, 2025.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

Leshem Choshen, Ryan Cotterell, Michael Y Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex

Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. [call for papers] the 2nd babylm challenge: Sample-efficient pretraining on a developmentally plausible corpus. *arXiv preprint arXiv:2404.06214*, 2024.

Eve V Clark and Marisa Casillas. First language acquisition. In *The Routledge handbook of linguistics*, pages 311–328. Routledge, 2015.

Max Coltheart. The mrc psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A*, 33(4):497–505, 1981.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. *arXiv preprint arXiv:2010.11929*, 2020.

Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6639–6648, 2019.

Martin Gerlach and Francesc Font-Clos. A standardized project gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126, 2020.

Jill Gilkerson, Jeffrey A Richards, Steven F Warren, Judith K Montgomery, Charles R Greenwood, D Kimbrough Oller, John HL Hansen, and Terrance D Paul. Mapping the early language environment using all-day recordings and automated analysis. *American journal of speech-language pathology*, 26(2):248–265, 2017.

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.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox, editors. *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, Miami, FL, USA, November 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.conll-babylm.0/.

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.

Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. Pytorch. *Programming with TensorFlow: solution for edge computing applications*, pages 87–104, 2021.

Anna A Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi Kumar, Setayesh Radkani, Thomas H Clark, Carina Kauf, Jennifer Hu, RT Pramod, Gabriel Grand, et al. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *arXiv preprint arXiv:2405.09605*, 2024.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

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

Alina Klerings, Christian Bartelt, and Aaron Mueller. Developmentally plausible multimodal language models are highly modular. In Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox, editors, *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 118–139, Miami, FL, USA, November 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.conll-babylm.10/.

Jing Yu Koh, Ruslan Salakhutdinov, and Daniel Fried. Grounding language models to images for multimodal inputs and outputs. In *International Conference on Machine Learning*, pages 17283–17300. PMLR, 2023.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and better understanding vision-language models: insights and future directions. In *Workshop on Responsibly Building the Next Generation of Multimodal Foundational Models*, 2024a.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *Advances in Neural Information Processing Systems*, 37:87874–87907, 2024b.

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.

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

Zongxia Li, Xiyang Wu, Hongyang Du, Huy Nghiem, and Guangyao Shi. Benchmark evaluations, applications, and challenges of large vision language models: A survey. *arXiv preprint arXiv:2501.02189*, 1, 2025.

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.

Pierre Lison and Jörg Tiedemann. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. 2016.

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

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.

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.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10437–10446, 2020.

Brian MacWhinney. *The CHILDES project: The database*, volume 2. Psychology Press, 2000.

Patrick E McKight and Julius Najab. Kruskal-wallis test. *The corsini encyclopedia of psychology*, pages 1–1, 2010.

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

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. Connecting

vision and language with localized narratives. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 647–664. Springer, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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.

Susan A Rose, Judith F Feldman, and Jeffery J Jankowski. A cognitive approach to the development of early language. *Child development*, 80(1):134–150, 2009.

Rohan Saha, Abrar Fahim, Alona Fyshe, and Alex Murphy. Exploring curriculum learning for vision-language tasks: A study on small-scale multimodal training. In Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox, editors, *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 65–81, Miami, FL, USA, November 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.conll-babylm.6/.

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.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3): 339–373, 2000.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019.

Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. B2t connection: Serving stability and performance in deep transformers. *arXiv preprint arXiv:2206.00330*, 2022.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5238–5248, 2022.

Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multi-

modal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212, 2021.

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.

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. *arXiv preprint arXiv:2205.14100*, 2022.

Jing Wang, Julie A Conder, David N Blitzer, and Svetlana V Shinkareva. Neural representation of abstract and concrete concepts: A meta-analysis of neuroimaging studies. *Human brain mapping*, 31(10):1459–1468, 2010.

Nan Wang and Qi Wang. Dynamic weighted gating for enhanced cross-modal interaction in multimodal sentiment analysis. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21(1):1–19, 2024.

Shaonan Wang, Jiajun Zhang, and Chengqing Zong. Learning multimodal word representation via dynamic fusion methods. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. Blimp: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020.

Alex Warstadt, Leshem Choshen, Aaron Mueller, Adina Williams, Ethan Wilcox, and Chengxu Zhuang. Call for papers – the BabyLM challenge: Sample-efficient pretraining on a developmentally plausible corpus. arXiv preprint arXiv:2301.11796, 2023. URL https://arxiv.org/abs/2301.11796.

Long-Fei Xie and Xu-Yao Zhang. Gate-fusion transformer for multimodal sentiment analysis. In *International Conference on Pattern Recognition and Artificial Intelligence*, pages 28–40. Springer, 2020.

Peng Xu, Xiatian Zhu, and David A Clifton. Multimodal learning with transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12113–12132, 2023.

Zihui Xue and Radu Marculescu. Dynamic multimodal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2575–2584, 2023.

Fei Zhao, Chengcui Zhang, and Baocheng Geng. Deep multimodal data fusion. *ACM computing surveys*, 56(9):1–36, 2024.

Chengxu Zhuang, Evelina Fedorenko, and Jacob Andreas. Visual grounding helps learn word meanings in low-data regimes. *arXiv preprint arXiv:2310.13257*, 2023.

Chengxu Zhuang, Evelina Fedorenko, and Jacob Andreas. Lexicon-level contrastive visual-grounding improves language modeling. *arXiv preprint arXiv:2403.14551*, 2024.

Appendix A

Dual Stream Transformer Hyperparameters

Model Hyperparameter	**Value**
Model Dimensions	
Model dimension (d_{model})	768
Hidden dimension	3072
Number of attention heads	8
Image encoder layers	5
Decoder layers	8
Vocabulary & Sequence	
Vocabulary size	50,260 (GPT-2 tokeniser (Radford et al., 2019))
Maximum sequence length	128
Special tokens	[PAD], [BOS], [EOS]
Activation & Regularisation	
Activation function	GELU (Hendrycks and Gimpel, 2016)
Dropout rate	0.1
Layer normalisation	Pre-layer norm
Layer norm epsilon	1e-5 (PyTorch default)
Input Dimensions	
DINOv2 embedding dimension	768
DINOv2 representation	*CLS* token only
Model Statistics	
Total parameters	$\sim$198.5M

Table A.1: The hyperparameters list for my dual stream transformer base model.

Appendix B

Experiments Summary

#	Model Architecture	Model Hyperparams	Training Config
1	Base model (§3.3)	Default[a]	Default[b]
Dynamic Gating			
2	Base model + no gate	Default[a]	Default[b]
3	Base model + soft gate per feature	Default[a]	Default[b]
4	Base model + soft gate per token	Default[a]	Default[b]
5	Base model + hard gate per feature	Default[a]	Default[b]
6	Base model + hard gate per token	Default[a]	Default[b]
Feature Representation			
7	Base model + FiLM on text	Default[a]	Default[b]
8	Base model + FiLM on image	Default[a]	Default[b]
9	Base model + FiLM on cross-attention	Default[a]	Default[b]
10	Base model + DyIntra on text	Default[a]	Default[b]
11	Base model + DyIntra on image	Default[a]	Default[b]
12	Base model + DyIntra on cross-attention	Default[a]	Default[b]
13	Base model + Channel Attention	Default[a]	Default[b]
Auxiliary Objectives			
14	Base model	Default[a]	Default[b] + CLIP (BS=128)
15	Base model	Default[a] + weight tying	Default[b] + LCG (BS=128)
Data Curriculum			
16	Base model	Default[a]	Text-only → image-caption[c]
17	Base model	Default[a]	Image-caption → text-only[d]
18	Base model	Default[a]	Non-uniform mix[e]
19	Base model	Default[a]	Uniform mix

[a]As in Table A.1 [b]As in Table 4.1 BS = batch size

[c]First 10 epochs text-only, next 10 epochs image-caption

[d]First 10 epochs image-caption, next 10 epochs text-only

[e]Image-caption and text-only data non-uniformly mixed in same batch

Table B.1: Summary of all the experiments I conduct in this work.

Appendix C

Design Choices for the Image Processing Pipeline

Despite using only global *CLS* image embeddings, I chose to implement an image encoder in my framework for the following reasons:

- Future compatibility: I aim to develop future iterations of this framework that address current limitations by using patch tokens instead of global image embeddings. For comparable results, I choose to use an encoder for the CLS token as well, which benefits from feed-forward and normalisation layers, but not self-attention. The image encoder outputs a non-linear adaptation of pretrained visual features and improves alignment with the text stream.
- Empirical performance: I experimented with three variants: (1) directly using linearly projected DINOv2 embeddings, (2) applying a 2-layer multi-layer-perceptron (MLP), and (3) using the transformer encoder. The encoder variant demonstrated superior performance across benchmarks, which can be attributed to the encoder's deeper transformation capacity. The benchmark scores for the three variants are available in table C.1.
- Computational efficiency: An alternative to the image encoder is to import and fully or partially unfreeze the external pretrained image encoder used in the BabyLM Challenge, *facebook/dino2-base*[1]. However, this would require processing the raw images through the entire encoder (86.6 million parameters) during training, which would significantly increase the computational costs for data loading, forward passes (and backward passes if unfrozen) and memory usage. This approach contradicts the constraints of the challenge, which advocates for the fair use of computational resources. In contrast, a customisable image encoder component taking as input pre-computed embeddings can be modified based on the user's computational constraints.

Table C.1 summarises the performance of the base model with different image processing pipelines, evaluated every 200,000 steps on BLiMP, BLiMP Supplement, EWoK, Winoground and VQA. As shown, the results on BLiMP, BLiMP Supplement and VQA validate the use of a transformer encoder,

[1] https://huggingface.co/facebook/dinov2-base

Model	Checkpoint	BLiMP	BLiMP S.	EWoK	Winoground	VQA
Base + No Encoder	200K	69.99 ± 0.17	54.03 ± 0.52	49.87 ± 0.57	51.74 ± 1.83	43.00 ± 0.31
	400K	73.21 ± 0.16	53.33 ± 0.62	49.93 ± 0.57	52.68 ± 1.83	46.54 ± 0.31
	600K	72.82 ± 0.16	52.89 ± 0.65	50.38 ± 0.57	52.95 ± 1.83	46.41 ± 0.31
	800K	73.40 ± 0.16	53.69 ± 0.64	50.16 ± 0.57	52.41 ± 1.83	46.52 ± 0.31
	1M	73.17 ± 0.16	53.62 ± 0.64	50.43 ± 0.57	52.14 ± 1.83	46.65 ± 0.31
	1.107M	74.29 ± 0.16	55.63 ± 0.58	50.18 ± 0.57	51.21 ± 1.83	45.02 ± 0.31
Base + MLP Encoder	200K	70.66 ± 0.17	57.42 ± 0.49	50.03 ± 0.57	52.01 ± 1.83	41.72 ± 0.31
	400K	73.47 ± 0.16	51.89 ± 0.66	49.96 ± 0.57	50.67 ± 1.83	48.41 ± 0.31
	600K	73.20 ± 0.16	52.49 ± 0.68	50.02 ± 0.57	52.01 ± 1.83	43.32 ± 0.31
	800K	74.06 ± 0.16	51.71 ± 0.67	50.47 ± 0.57	52.28 ± 1.83	48.18 ± 0.31
	1M	73.71 ± 0.16	50.56 ± 0.67	50.21 ± 0.57	52.55 ± 1.83	48.54 ± 0.31
	1.107M	74.35 ± 0.16	55.38 ± 0.60	50.21 ± 0.57	50.27 ± 1.83	49.83 ± 0.31
Base + Transformer Encoder	200K	69.97 ± 0.17	55.02 ± 0.54	49.83 ± 0.57	51.07 ± 1.83	34.11 ± 0.32
	400K	73.22 ± 0.16	51.88 ± 0.66	50.13 ± 0.57	50.94 ± 1.83	47.82 ± 0.31
	600K	73.47 ± 0.16	53.08 ± 0.62	50.74 ± 0.57	51.61 ± 1.83	48.36 ± 0.31
	800K	74.18 ± 0.16	52.69 ± 0.62	50.69 ± 0.57	52.41 ± 1.83	51.2 ± 0.31
	1M	74.34 ± 0.16	54.00 ± 0.61	50.82 ± 0.57	52.41 ± 1.83	51.2 ± 0.31
	1.107M	75.53 ± 0.16	55.71 ± 0.57	50.41 ± 0.57	51.74 ± 1.83	50.02 ± 0.31

Table C.1: The performance of the base model with different image processing pipelines. The models are evaluated every 200,000 steps on BLiMP, BLiMP Supplement, EWoK, Winoground and VQA.

for which the base model achieves the best scores. However, since a single image embedding cannot benefit from the self-attention mechanism, an MLP encoder suffices if computational resources are a constraint, achieving competitive performance.Besides the superior performance, the motivation for using a transformer encoder in this work was to enable a direct performance comparison with future iterations of the framework using patch-token embeddings.

Appendix D

Explanation for Base Model Performance Oscillation on BLiMP Supplement

As discussed in 5.5, the performance of the base model significantly oscillates between text-only and image-caption epochs for the *subject-auxiliary inversion* and *turn taking* substask of BLiMP Supplement. I thus investigate the log probability scores of my base model for each subtask example at checkpoints 500,000 (text-only epoch) and 550,000 (image-caption epoch), for which the former model is incorrect and the latter is correct. These two checkpoints present the highest difference in BLiMP Supplement scores (7.54%). I make the following observations:

1. For the *subject-auxiliary inversion*, 68.2% of the examples for which the model at checkpoint 500,000 (text-only epoch) is incorrect and the model at checkpoint 550,000 (image-caption epoch) is correct have the correct sentence of the pair starting with "Is" followed by a noun phrase. For example, pairs such as ("Is the host expecting an award-winning director that hasn't finished dressing yet?", "Hasn't the host is expecting an award-winning director that finished dressing yet?"). This contrasts with the distribution of the task, where 31.1% of the pairs have the correct sentence starting with "Is" followed by a noun phrase. I theorise that the Localized Narratives dataset supports the model at checkpoint 550,000 (image-caption epoch) in choosing the "Is" followed by noun phrase sentences with higher probability, which happen to be the correct sentences in these pairs. That is because there are 706,251 constructions of the form "there is" followed by a noun phrase in the Localized Narratives dataset. I hypothesise that as a result, my model learns that the pattern "is" followed by a noun phrase is more likely during the image-caption epochs.

2. For the *turn taking* subtask, even if the model at checkpoint 550,000 (image-caption epoch) chooses the correct sentence more often, it does so with little confidence. For most examples for which checkpoint 550,000 (image-caption epoch) is correct and checkpoint 500,000 (text-only epoch) is not, the log probability difference between the correct and incorrect sentence of the former checkpoint is less than 2 points. To put this in context, the log probability scores range between -89 and -156, for which 2 points represent 0.013% to 0.0225%. There is no

noticeable pattern in the training data that can motivate the model's better performance during image-caption epochs on the the *turn taking* subtask. I conclude that this behaviour requires further investigation which I leave for future work.

Appendix E

BabyLM Challenge Vision Track Training Dataset

Pretraining Dataset	Description	# Words	# Images
Localized Narratives (Pont-Tuset et al., 2020)	image-caption	27M	0.6M
Conceptual Captions 3M (Sharma et al., 2018)	image-caption	23M	2.3M
CHILDES (MacWhinney, 2000)	child-directed speech	15M	-
British National Corpus (BNC), dialogue portion	dialogue	4M	-
Project Gutenberg, children's stories (Gerlach and Font-Clos, 2020)	written English	13M	-
OpenSubtitles (Lison and Tiedemann, 2016)	movie subtitles	10M	-
Simple English Wikipedia	written English	7M	-
Switchboard Dialog Act (Stolcke et al., 2000)	dialogue	<1M	-
Total		100M	2.9M

Table E.1: Data sources and corresponding word and image approximate counts in the multimodal pretraining dataset of the BabyLM Challenge. Adapted from (Choshen et al., 2024).

Appendix F

Extra Plots

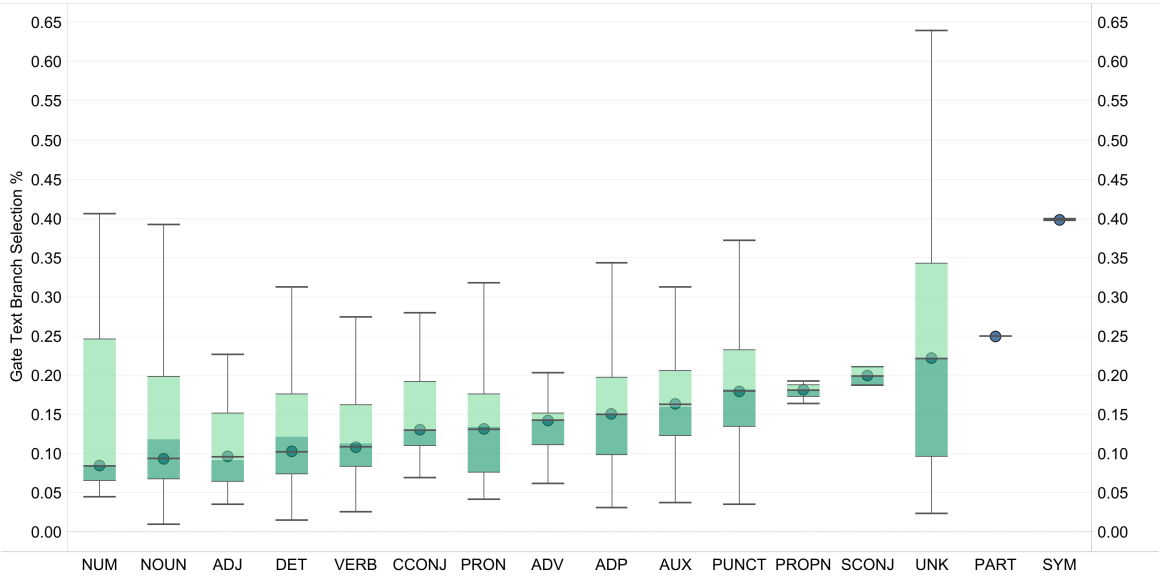


Figure F.1: *Hard gate per feature* gate selection per part-of-speech.