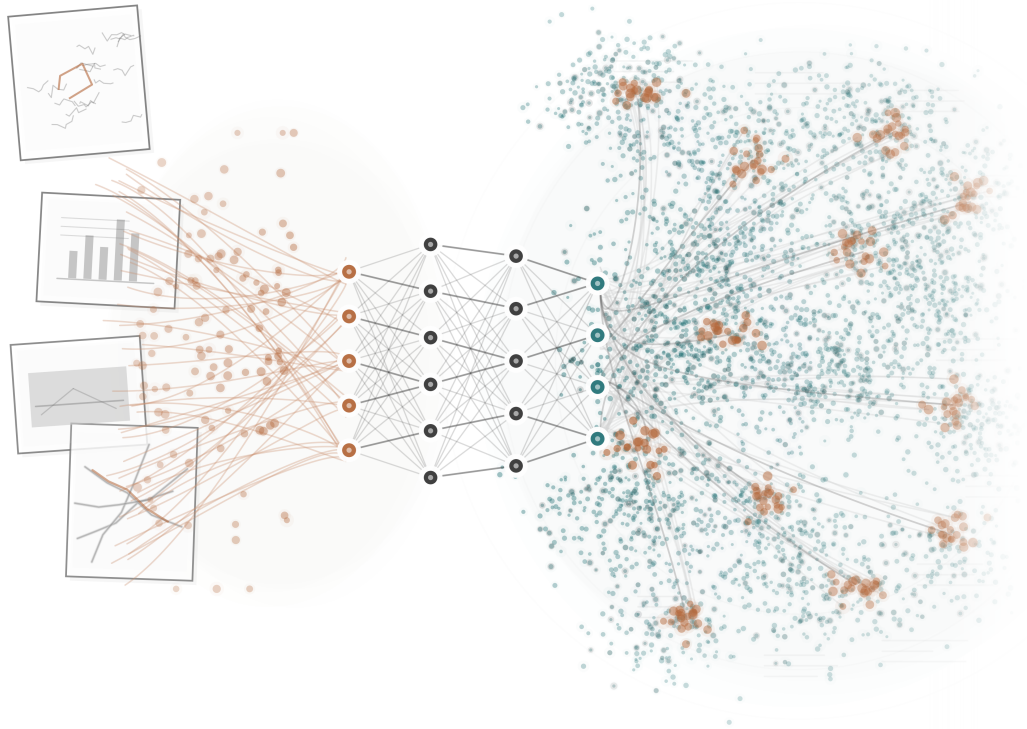




CHALMERS
UNIVERSITY OF TECHNOLOGY



Multimodal Retrieval-Augmented Generation via Embedding Space Alignment

Bridging image and text representations for multimodal document retrieval

Master's thesis in Complex Adaptive Systems

OSKAR ALBERS

DEPARTMENT OF PHYSICS

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

MASTER'S THESIS 2026

Multimodal Retrieval Augmented Generation via Embedding Space Alignment

Bridging image and text representations for multimodal document
retrieval

OSKAR ALBERS



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Physics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026

Multimodal Retrieval Augmented Generation via Embedding Space Alignment
Bridging image and text representations for multimodal document retrieval
OSKAR ALBERS

© OSKAR ALBERS, 2026.

Industrial Supervisor: Adam Ek, AI Sweden
Academic Supervisor: Daniel Midtvedt, Department of Physics
Examiner: Giovanni Volpe, Department of Physics

Master's Thesis 2026
Department of Physics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Conceptual illustration of image embeddings from documents being mapped through a learned neural bridge into a shared text retrieval space. Created by the author using Python and Matplotlib.

Typeset in L^AT_EX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Multimodal Retrieval-Augmented Generation via Embedding Space Alignment
Bridging Image and Text Representations for Multimodal Document Retrieval
OSKAR ALBERS
Department of Physics
Chalmers University of Technology

Abstract

Retrieval-augmented generation (RAG) systems answer questions by first retrieving supporting evidence and then generating a response conditioned on that evidence. In visually rich document collections, such as Swedish municipal PDF archives, relevant information is often spread across text, maps, diagrams, and photographs. Many deployed pipelines are built around strong text embedding models like E5, and fully migrating to a new multimodal retriever can be costly and disruptive.

This thesis investigates whether visual retrieval can be added to an existing E5-based text pipeline by aligning image representations to its embedding space, instead of re-embedding the corpus with a unified multimodal model. Concretely, a neural “bridge” is trained to project SigLIP image embeddings into the E5 text space, using a staged procedure (MAPS) that gradually shapes the target space, leverages paired question–image and question–text examples, and learns calibrated joint ranking between images and text. A visual-question-answering style benchmark is constructed from Swedish municipal documents, where each natural-language question is paired with the images or text passages from which it is answerable.

On a corpus of 235,653 documents, 901,571 text chunks, and 427,843 images, naïve combinations of E5 and SigLIP scores fail to retrieve relevant images, yielding essentially no high-ranked image hits. The learned bridge, in contrast, raises image-only retrieval performance from single-digit to high-20% top-ranked hit rates and achieves comparable gains in a joint image–text setting, while largely preserving text retrieval beyond the very top rank. These results suggest that embedding-space alignment could be a practical strategy for extending text-centric RAG infrastructure with visual search over document images, without replacing the underlying text encoder or index.

Keywords: Retrieval-Augmented Generation, Multimodal Retrieval, Embedding Space Alignment, Visual Question Answering, Vec2Vec.

Acknowledgements

I would like to express my gratitude to my industrial supervisor, Adam Ek, and to AI Sweden for making this thesis possible. The opportunity to work in a practically relevant setting, with access to real-world data, infrastructure, and technical support, has been essential for carrying out this project.

I would also like to thank my academic supervisor, Daniel Midtvedt, for his support and help with the thesis process, and my examiner, Giovanni Volpe, for overseeing the project from the university side.

Oskar Albers, Gothenburg, May 2026

Declaration of AI Usage

Generative AI tools have been used to support brainstorming, proofreading, language refinement, and programming assistance during the preparation of this thesis. The experimental design, analyses, interpretations, and final decisions were made by the author, who takes full responsibility for the content of the thesis.

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis, listed in alphabetical order:

ANN	Approximate Nearest Neighbour
API	Application Programming Interface
CE	Cross Entropy
CLIP	Contrastive Language–Image Pre-training
E5	Multilingual-E5-Large (text embedding model)
EWC	Elastic Weight Consolidation
FFN	Feedforward Network
GELU	Gaussian Error Linear Unit
GPU	Graphics Processing Unit
HNSW	Hierarchical Navigable Small World
InfoNCE	Information Noise Contrastive Estimation
JSONL	JSON Lines
LLM	Large Language Model
MAPS	Multimodal Alignment with Paired Supervision
MLP	Multilayer Perceptron
OCR	Optical Character Recognition
PCA	Principal Component Analysis
PDF	Portable Document Format
QA	Question Answering
RAG	Retrieval-Augmented Generation
ReLU	Rectified Linear Unit
SigLIP	Sigmoid Loss for Language–Image Pre-training
VLM	Vision–Language Model
VQA	Visual Question Answering
VRAM	Video Random Access Memory
VRD	Visually Rich Document

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables that have been used throughout this thesis.

Indices

i	Index for an image, text chunk, candidate item, or embedding vector
j	Index for a candidate item in a ranked list or score matrix
c	Index for an image routing category
t	Index for a token position
B	Batch size
K	Number of top-ranked results considered in retrieval metrics

Sets

$\mathcal{X}$	Domain of input objects, such as text passages or images
$\mathcal{D}$	Document or retrieval collection
$\mathcal{Q}$	Set of evaluation queries
$\mathcal{P}_c^+$	Set of positive routing prompts for category c
$\mathcal{P}_c^-$	Set of negative routing prompts for category c
$C_K(q)$	Top- K retrieved evidence set for query q
$L_K(q)$	Ordered top- K ranked list returned for query q

Embeddings and Models

$\mathbf{e}_x$	Dense embedding vector for object x
$\tilde{\mathbf{e}}_x$	Mean-centered and L2-normalized E5 embedding vector

$\mathbf{e}_i^{s,\text{img}}$	SigLIP source-space image embedding for image i
$\mathbf{e}_i^{t,\text{text}}$	E5 target-space text embedding for text item i
$\mathbf{q}$	Query embedding
$\tilde{\mathbf{q}}$	Centered E5 query embedding
$\mathbf{p}$	SigLIP text embedding of a routing prompt
f_θ	Parameterized embedding model with parameters θ
f_θ^q	Query encoder
f_θ^d	Document or passage encoder
g_ϕ	Bridge model mapping with parameters ϕ
d	Embedding space dimensionality
d_s	Source embedding dimensionality, SigLIP: 1152
d_t	Target embedding dimensionality, E5: 1024

Retrieval and Ranking Variables

i^*	Ground-truth relevant item for a query
r_q	Rank of the ground-truth item for query q
$\mathbf{Q}$	Matrix of query embeddings in a training batch
$\mathbf{C}$	Matrix of candidate embeddings in a training batch
$\mathbf{S}$	Score matrix between queries and candidates
S_{ij}	Score between query i and candidate j
$s_{i,c}$	Raw routing score for image i and category c
$\hat{s}_{i,c}$	Z-score calibrated routing score for image i and category c
μ_c, σ_c	Mean and standard deviation of raw routing scores for category c
ϵ	Small constant for numerical stability
$z_{\min}$	Minimum Z-score threshold for routing, set to -1
δ	Minimum routing margin threshold, set to 0.25

Training and Calibration Parameters

τ	Temperature parameter used in contrastive or ranking losses
α	Learned multiplicative image score scale in Stage 4 calibration
β	Learned additive image score bias in Stage 4 calibration
γ	Collapse regulariser weight in Stage 4

λ	Generic regularisation weight
λ_{disp}	Text displacement penalty weight
λ_{α}	Regularisation weight for the image scale parameter
λ_{β}	Regularisation weight for the image bias parameter
λ_1	Laplacian regulariser weight in Stage 3
λ_2	EWC penalty weight in Stage 3

Functions and Operators

$\cos(\cdot, \cdot)$	Cosine similarity between two vectors
top- K	Operator returning the K highest-scoring items
Hits@ K	Fraction of queries where the ground-truth item appears in the top- K ranking
$\mathbf{1}[\cdot]$	Indicator function
$\ \cdot\ _2$	Euclidean norm

Contents

Declaration of AI Usage	ix
List of Acronyms	x
Nomenclature	xiii
List of Figures	xxi
List of Tables	xxiii
1 Introduction	1
1.1 Background and motivation	1
1.2 Thesis aim and scope	2
1.2.1 Research aim and questions	3
1.2.2 Scope limitations	3
1.2.3 Expected contributions	3
1.3 Outline of the thesis	3
2 Theory	5
2.1 Neural Embeddings and Similarity	5
2.1.1 From Discrete Objects to Continuous Vectors	5
2.1.2 Embedding Models	7
2.1.3 Similarity and Distance Measures	7
2.2 Large Language Models and the Need for Retrieval	9
2.2.1 The Transformer Architecture	9
2.2.2 Parametric Knowledge and Its Limitations	10
2.3 Dense Retrieval and Retrieval-Augmented Generation	10
2.3.1 Dense Retrieval and Approximate Indexing	11
2.3.2 Retrieval-Conditioned Generation	12
2.4 Multimodal and Visual Document Embeddings	12
2.4.1 Visual Information in Documents	13
2.4.2 Vision–Language Embedding Models	13
2.5 Embedding Space Alignment	14
2.5.1 Cross-Modal Incompatibility	14
2.5.2 Embedding-Space Translation	15
2.5.3 Extending Embedding-Space Translation to Multimodal Re- trieval	16

2.6	Evaluation of Retrieval and VQA-style Tasks	17
2.6.1	Retrieval Metrics	17
2.6.2	VQA-style Evaluation	18
3	Methods	19
3.1	System Overview	19
3.2	Document Corpus and Indexing	19
3.2.1	Corpus	20
3.2.2	Preprocessing	20
3.2.3	Vector Indexing	21
3.3	VQA Dataset Construction	21
3.3.1	Image Filtering	21
3.3.1.1	Category Label Noise Estimation	22
3.3.2	VQA Triplet Generation	23
3.3.2.1	Gatekeeper	23
3.3.2.2	Generator	24
3.3.2.3	Judge	24
3.3.3	Text QA Generation	25
3.3.3.1	Generator	25
3.3.3.2	Judge	25
3.3.4	Gold Filtering	26
3.4	Baseline Systems	26
3.4.1	SigLIP Image-Only	27
3.4.2	Naive Joint Retrieval	27
3.4.3	SigLIP Multimodal Baseline	27
3.5	Bridge Model	27
3.5.1	Architecture	28
3.6	MAPS Training Pipeline	28
3.6.1	Stage 0: Target Space Shaping	30
3.6.2	Stage 1: Text-to-Text Pretraining	30
3.6.3	Stage 2: Anchor Fit	31
3.6.4	Stage 3: Corpus Extension	31
3.6.4.1	Phase 1: EWC Anchor	32
3.6.4.2	Phase 2: Context-Grounded Contrastive Learning	32
3.6.4.2.1	Pseudo-label construction.	33
3.6.4.2.2	Three-term loss.	33
3.6.5	Stage 4: Unified Multimodal Ranking	34
3.6.5.1	Problem Formulation	35
3.6.5.2	Mixed Candidate Ranking Loss	35
3.6.5.3	Auxiliary Losses	36
3.6.5.4	Score Calibration	36
3.7	Experimental Design	37
3.7.1	Evaluation Conditions	37
3.7.2	Evaluation Protocol	37
3.7.3	Ablation Study	38
3.8	Compute	38

4	Results	39
4.1	Dataset Construction Results	39
4.1.1	Dataset Counts	39
4.1.2	Image Routing Noise	40
4.2	Retrieval Results	41
4.2.1	Baseline Score Incompatibility	41
4.2.2	Image-Only Retrieval	42
4.2.3	Joint Image-Text Retrieval	42
4.2.4	Text Retrieval and Displacement	43
4.3	Category and Query Type Analysis	43
4.3.1	Category-Level Performance	43
4.3.2	Query Type Analysis	44
4.4	Ablation Study	45
4.4.1	Bridge Capacity	45
4.4.2	Joint Retrieval and Text Displacement	45
5	Conclusion	49
5.1	Summary of Findings	49
5.2	Contributions	50
5.3	Practical Implications	50
5.4	Limitations	51
5.5	Future Work	52
5.6	Final Remarks	53
	Bibliography	55
A	Appendix A: Generation Prompts	I
A.1	VQA Gatekeeper Prompt	I
A.2	VQA Generator Prompts	I
A.3	VQA Judge Prompt	III
A.4	Text QA Generator Prompt	IV
A.5	Text QA Judge Prompt	V
B	Appendix B: Image Routing Prompts	VII
C	Hyperparameters	XI

List of Figures

2.1	Toy PCA projection of E5 embeddings from football, music, and <i>The Lord of the Rings</i> . Labels mark representative terms, while faint points show additional related terms. Distances are illustrative and should not be interpreted as exact high-dimensional embedding distances.	6
2.2	Illustrative overview of a retrieval-augmented generation (RAG) pipeline. A query q is used to retrieve a set of relevant items $C_K(q)$ from an embedding space. The retrieved evidence is combined with the query to form an augmented context, which is provided to a language model to generate the answer.	11
3.1	Overview of the end-to-end multimodal retrieval pipeline. The document corpus is split into text chunks and images, which are processed along two parallel branches. Text chunks are encoded with E5 and stored in an E5 text index, while images are encoded with SigLIP and mapped by the bridge model g_ϕ into an E5-compatible image index. At query time, a single E5 query embedding scores both indexes, the results are jointly ranked to form the retrieved evidence set $C_K(q)$, and the selected evidence is passed to the LLM to generate the answer $a_{1:T}$	20
3.2	The three-stage VQA generation pipeline. Each image is first assessed by a gatekeeper, then passed to a generator that produces a question-answer-explanation triplet, and finally evaluated by a judge that scores factual grounding.	23
3.3	Architecture of the bridge model used to project SigLIP image embeddings into the E5 text embedding space. Panel (1) shows the internal structure of a residual BridgeBlock, where the input is processed by a pre-normalized feed-forward network and added back through a residual skip connection. Panel (2) shows the full bridge model: a SigLIP image vector $\mathbf{v}_i \in \mathbb{R}^{1152}$ is mapped to 1024 dimensions by an input projection, passed through a stack of BridgeBlocks, and L2-normalized to produce an E5-compatible representation $g_\phi(\mathbf{v}_i) \in \mathbb{R}^{1024}$ for joint image-text retrieval.	28

3.4	MAPS training pipeline. SigLIP image vectors and E5 text/query vectors are brought into a shared retrieval setting through five stages: E5 target-space shaping, text-to-text bridge pretraining, VQA-based image–query anchoring, corpus-level pseudo-query extension, and unified image–text ranking. The final stage learns the image score calibration parameters used when projected images and E5 text chunks compete in the same joint retrieval index.	29
-----	--	----

List of Tables

3.1	Image routing categories and their disposition in the dataset construction pipeline.	22
3.2	Judge scoring rubric for factual grounding of generated triplets. . . .	25
4.1	Dataset counts after generation, judging, filtering, and splitting. The generated count is not reported for Text QA because non-gold records were discarded and not tracked during generation.	40
4.2	Distribution of gold VQA records across the three visual categories and two query types.	40
4.3	Estimated image routing noise by category. Noise rate is the fraction of sampled images judged to belong to a different category. 95% Wilson score confidence intervals computed from $n = 50$ samples per category; the Wilson interval is used because the normal-approximation interval is degenerate at the zero misroutes observed for maps	40
4.4	Text and image score distributions for the baseline joint conditions. Score means are computed over all queries and candidates in the evaluation pool.	41
4.5	Image-only retrieval results. Both conditions rank over all images in the dataset. The gray row is the bridge condition. Bold denotes the best value per column.	42
4.6	Joint image retrieval results. All conditions rank over all images and text chunks in the dataset simultaneously. The gray row indicates the bridge condition where images and text are scored in a compatible space. Bold denotes the best value per column.	42
4.7	Text retrieval on the held-out 500-record text QA evaluation set, shown as deltas (Δ) from the centered E5 text-only baseline. $\Delta = \text{condition} - \text{centered baseline}$: negative values indicate a decrease relative to the centered baseline (displacement), positive values an improvement. The text-joint condition additionally includes bridge-projected image vectors. The gray row indicates the bridge performance.	43
4.8	Category-level image retrieval results for the SigLIP image-only baseline (S) and bridge image-only condition (B). Bridge columns are shaded. Bold denotes the best value per column.	44
4.9	Image retrieval by query type for the SigLIP image-only baseline (S) and bridge image-only condition (B). Bridge columns are shaded. Bold denotes the best value per column.	44

4.10	Bridge capacity ablation in the image-only setting after Stage 4 calibration. Parameter counts exclude the frozen SigLIP and E5 encoders. The gray row is the final bridge used in all other experiments. Bold denotes the best value per column.	45
4.11	Bridge capacity ablation under joint retrieval after Stage 4 calibration. Images compete directly with all text chunks in the dataset. The gray row is the final bridge used in this thesis. Bold denotes the best value per column.	46
4.12	Text retrieval displacement under joint retrieval for each bridge capacity, shown as deltas (Δ) from the centered E5 text-only baseline. $\Delta = \text{condition} - \text{baseline}$: negative values indicate displacement of text by bridge-projected images. The dark gray row is the reference baseline (raw values), and the light gray row is the final 4-block bridge used in this thesis.	46
C.1	Hyperparameter settings for the MAPS training pipeline, general settings and Stages 1–2.	XI
C.2	Hyperparameter settings for Stage 3 of the MAPS training pipeline. .	XII
C.3	Hyperparameter settings for Stage 4 of the MAPS training pipeline. .	XIII

1

Introduction

1.1 Background and motivation

Large language models (LLMs) have quickly become part of everyday life. Through applications such as ChatGPT and Gemini, they are increasingly used to search for and access information, assist with writing, support coding tasks, and help users navigate large volumes of digital content [1], [2]. For many users, this has made LLMs a convenient and flexible interface for information access, allowing large amounts of content to be explored through natural language rather than conventional search alone. At the same time, their growing use has made their limitations more visible, particularly in cases where fluent and convincing answers turn out to be incorrect or unsupported, a phenomenon known as hallucination [3], [4]. This remains a persistent challenge in open-domain question answering and creates a clear risk when LLMs are used not only for convenience, but as tools for reliable information access and decision-making.

Retrieval-augmented generation (RAG) addresses this limitation by combining a language model with a retrieval component that fetches relevant external information before response generation [5], [6], [7]. Rather than relying solely on internal parametric knowledge, which is fixed after training and may be incomplete or outdated, RAG grounds outputs in evidence retrieved from external knowledge sources [5]. This makes the retrieval stage central to the reliability of the overall system: semantic mismatch, incomplete indexing, or poor ranking can prevent relevant evidence from reaching the generator, limiting the quality of the final answer even when the language model itself is strong [6], [8].

RAG was initially developed in text-only settings, retrieving passages from textual corpora to ground language generation [5]. Multimodal RAG extends this framework by allowing retrieval and generation to make use of non-textual evidence such as images, tables, diagrams, audio, and video [9], [10]. This development has been enabled by progress in vision-language models such as CLIP, Flamingo, and LLaVA, which support cross-modal representation and reasoning over text and visual inputs [11], [12], [13]. However, compared with text-only RAG pipelines, multimodal RAG remains less standardized. Recent surveys identify cross-modal retrieval, alignment, fusion, evaluation, and robustness as open challenges for building reliable multimodal RAG systems [9], [10].

The challenge is particularly pronounced in document collections such as PDFs, where essential information is distributed across textual and visual elements, including figures, tables, diagrams, layout structures, maps, and rendered page ge-

ometry [14], [15], [16]. Conventional text extraction recovers linguistic content, but often discards visual layout, spatial relationships, graphical marks, and embedded visual evidence that may be necessary for answering user queries. For example, a citizen asking where parking is permitted on a specific street may need both the textual parking ordinance and a local planning map showing the regulated segment, neither of which is fully captured by text extraction alone.

Recent vision-language retrievers such as ColPali address this limitation by embedding rendered document pages directly and have shown strong performance on visually rich document retrieval benchmarks such as ViDoRe [14]. These results demonstrate the importance of visual information in document retrieval, but they also highlight a practical challenge: using a new vision-language retriever often requires re-indexing the corpus and replacing existing text-oriented retrieval infrastructure.

This thesis investigates this challenge in a concrete, practically motivated setting: a collection of Swedish municipal PDF documents. This corpus is a representative case of real-world mixed-media archives: it contains maps, planning diagrams, regulatory figures, tables, and photographs whose informational content cannot always be recovered by text extraction alone, yet these documents must be searchable by citizens and administrators using natural language queries. Extending an existing text retrieval pipeline to handle this visual content without rebuilding the entire index infrastructure is therefore both a practical necessity and an open research problem.

1.2 Thesis aim and scope

A complete re-embedding of a large document corpus using a unified vision-language model is computationally expensive and requires replacing established text retrieval infrastructure. This is especially relevant when the existing pipeline already uses a strong retrieval-oriented text embedding model, such as E5, whose embeddings are designed for dense retrieval over textual corpora [17]. In many organisational settings it is therefore desirable to improve multimodal retrieval capabilities while keeping the underlying text index and text encoder fixed.

This thesis instead investigates whether the visual gap can be bridged more efficiently through vector-space alignment: projecting image embeddings from an independently trained vision-language model into an existing text embedding space, such as E5, BERT, or RoBERTa, thereby enabling joint retrieval without modifying the underlying text index.

This Vec2Vec-style approach is motivated by recent work showing that embedding spaces trained independently can sometimes be related through learned mappings, suggesting that useful structure may be transferred across representation spaces [18], [19], [20]. In this thesis, we explore the Vec2Vec alignment approach in a harder cross-modal setting, where SigLIP image embeddings are projected into an E5 text embedding space and evaluated against unified multimodal embedding baselines.

1.2.1 Research aim and questions

The central research aim is to determine whether Vec2Vec alignment constitutes an effective method for adapting text retrieval pipelines to multimodal documents, compared empirically against full multimodal embedding approaches. Specifically, this work addresses two research questions:

1. Can Vec2Vec mappings achieve retrieval-compatible alignment between independently trained text and image embedding spaces?
2. Which approach - Vec2Vec alignment or unified multimodal embeddings, in this thesis: SigLIP - demonstrates better retrieval effectiveness on mixed-media document collections?

These questions are addressed empirically through the methods and evaluation protocol described in Chapter 3 and the results reported in Chapter 4.

1.2.2 Scope limitations

The investigation remains strictly confined to retrieval-stage modifications within existing text-oriented pipelines. The study does not extend to developing complete end-to-end multimodal RAG systems, nor does it address multimodal fusion, reasoning, or generation strategies during response synthesis. The focus rests exclusively on retrieval methods that maintain compatibility with established text-based architectures, where the text encoder and text index are kept fixed and visual support is added through a learned mapping.

1.2.3 Expected contributions

The work offers three principal contributions. First, it introduces MAPS (Multimodal Alignment with Paired Supervision), a staged training pipeline for cross-modal embedding alignment under scarce supervision, combining InfoNCE anchor fitting, an EWC-stabilised consistency loss against a frozen Stage 2 reference model, confidence-gated pseudo-label contrastive training, and joint score calibration. Second, it contributes a domain-specific multimodal evaluation dataset of 22,509 gold visual question answering pairs derived from Swedish municipal PDF documents, constructed through an automated generation and LLM-as-judge pipeline. Third, it provides a systematic empirical comparison of Vec2Vec alignment against unified multimodal embedding approaches on a large real-world mixed-media corpus, offering evidence-based guidance on when and how vector-space alignment constitutes a viable strategy for extending text retrieval pipelines to multimodal documents.

1.3 Outline of the thesis

The remainder of this thesis is organized as follows. Chapter 2 provides the theoretical foundations and related work on retrieval-augmented generation, multimodal retrieval systems, and vector-space embedding alignment methods. Chapter 3 presents

the dataset construction, retrieval pipeline architecture, representation strategies, and experimental design. Chapter 4 reports the empirical results and analyzes retrieval performance across the proposed approaches. Finally, Chapter 5 summarizes the key findings, discusses limitations, and outlines directions for future research.

2

Theory

This chapter introduces the theoretical concepts needed to understand the multi-modal retrieval-augmented generation (RAG) framework studied in this thesis. The central problem is how to retrieve relevant evidence from a document collection where information is distributed across both text and visual content. In a standard text-based RAG system, queries and text passages are embedded into a shared vector space and retrieved by geometric similarity. However, this assumption breaks down when visual elements such as maps, figures, and diagrams must be retrieved together with text, since they are not encoded into the same space by default.

The chapter begins by introducing neural embeddings and the similarity measures used to compare them. It then introduces the transformer-based language models that generate answers in RAG systems and shows why they must also read from a specific document collection, instead of relying only on what is stored in their parameters. Building on this, dense retrieval and the RAG pipeline are formalized. The chapter then turns to visual document embeddings and the challenge of combining independently trained text and image encoders in a single retrieval index. It then treats embedding-space alignment: why independently trained spaces are geometrically incompatible, how embedding-space translation establishes that a learned mapping can bridge them, and why extending this idea to multimodal retrieval is qualitatively more demanding than the text-to-text precedent. The chapter concludes with the retrieval metrics and VQA-style evaluation formulation used to assess retrieval quality.

2.1 Neural Embeddings and Similarity

In this section, we introduce the concept of neural embeddings and how similarity in embedding spaces is quantified. The aim is to define the mathematical objects and notation that will be used throughout the thesis, and to provide an intuitive understanding of why vector representations are suitable for representing both text and images in the multimodal RAG setting.

2.1.1 From Discrete Objects to Continuous Vectors

Many of the objects considered in this thesis, such as document passages, questions, figures, maps, and rendered pages, cannot be compared directly in their raw form. Text is represented as sequences of discrete tokens, while images are represented as arrays of pixel values. These raw representations are poorly suited to seman-

tic retrieval, where the goal is to compare objects by meaning or function rather than by their surface form. A retrieval system therefore requires representations in which objects can be geometrically compared based on their semantic or functional properties [21].

Classical text encodings provide simple ways to convert text into numerical form. One-hot encoding represents each distinct symbol as a high-dimensional binary vector with a single nonzero entry. Bag-of-words represents a document as a vocabulary-indexed vector, recording term presence, counts, or weights while ignoring word order. Such representations are sparse, high-dimensional, and depend heavily on exact lexical overlap [21]. As a result, two passages that describe a similar concept using different words, such as “parking is prohibited on this street” and “no stopping zone”, may appear less related than desired.

Modern retrieval systems instead rely on dense vector representations. Each object x from a domain $\mathcal{X}$, such as a sentence, document, or image, is mapped to a continuous vector $\mathbf{e}_x \in \mathbb{R}^d$. The aim is for geometric relations between vectors to reflect semantic or functional relations between the underlying objects: related objects should be mapped to nearby points in the embedding space, while unrelated objects should lie farther apart. This property is what makes dense embeddings useful for semantic search, since retrieval can depend on meaning rather than only on lexical overlap [21], [22].

To make this intuition concrete, Figure 2.1 shows a toy visualization of E5 embeddings for words and short phrases from three semantic domains: FOOTBALL, MUSIC, and THE LORD OF THE RINGS. The embeddings are projected to two dimensions using principal component analysis (PCA). A few representative terms are shown as labels, while additional related terms are shown as faint points. Since PCA is only a linear low-dimensional projection, the figure should be interpreted as an illustration of semantic neighborhoods rather than an exact map of distances in the original high-dimensional embedding space [17], [21].

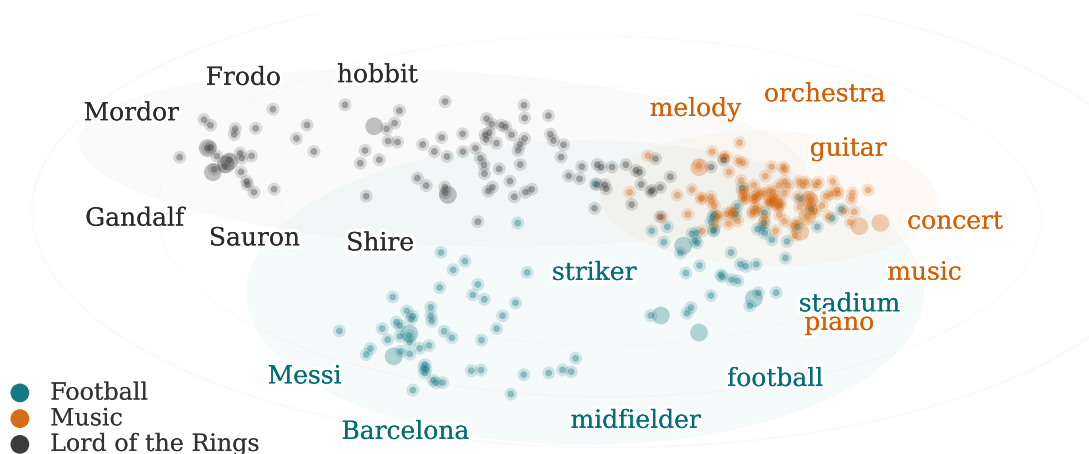


Figure 2.1: Toy PCA projection of E5 embeddings from football, music, and *The Lord of the Rings*. Labels mark representative terms, while faint points show additional related terms. Distances are illustrative and should not be interpreted as exact high-dimensional embedding distances.

The projection nevertheless illustrates the central intuition used throughout this thesis: embeddings make it possible to treat retrieval as geometric search. Although the figure shows individual words and short phrases, the same idea extends to passages, queries, images, and rendered pages. The concrete encoders used for these representations are introduced in the next subsection.

2.1.2 Embedding Models

An *embedding model* is a parameterized function $f_\theta : \mathcal{X} \rightarrow \mathbb{R}^d$ that maps an input object $x \in \mathcal{X}$, such as a text passage, image, or rendered page, to a dense vector $\mathbf{e}_x = f_\theta(x)$ in a fixed-dimensional embedding space. In modern retrieval systems, this function is typically implemented as a neural encoder trained so that similarity between output vectors correlates with semantic relevance [17], [23]. In this thesis, the embedding models are treated as fixed pretrained encoders: their parameters θ are not updated, and their output vectors are used directly for indexing, retrieval, and alignment.

For text retrieval, transformer-based encoders map a query or passage to a single vector representation. Retrieval-oriented models such as E5 are trained to produce text embeddings that transfer to tasks such as semantic search, clustering, and classification [17]. This single-vector formulation is particularly useful for dense retrieval, since both queries and candidate passages can be embedded independently and compared efficiently by vector similarity.

For visual content, images or rendered document pages can be embedded using vision encoders or vision–language models. Vision Transformer models process images as sequences of patches, allowing transformer architectures to be applied to visual inputs [24]. Vision–language models such as CLIP, ALIGN, and SigLIP use paired image–text supervision to learn representations in which matching images and captions receive higher similarity than mismatched pairs [11], [25], [26]. SigLIP follows this general image–text pretraining paradigm, but uses a pairwise sigmoid loss rather than the standard softmax contrastive loss [26].

In multimodal retrieval, text and visual content may therefore be represented by different encoders. A text encoder f_θ^t maps text inputs to embeddings $\mathbf{e}^t \in \mathbb{R}^{d_t}$, while a vision encoder f_θ^v maps visual inputs to embeddings $\mathbf{e}^v \in \mathbb{R}^{d_v}$. If the encoders are trained jointly, their outputs can be directly comparable within a shared vision–language embedding space. If they are trained independently, however, their output spaces are generally not geometrically compatible. This distinction is central to the alignment problem studied in this thesis.

The next step is to define how embeddings are compared once they have been produced. This requires similarity and distance measures, which determine how retrieved items are ranked in the embedding space.

2.1.3 Similarity and Distance Measures

Given two embeddings $\mathbf{e}_x, \mathbf{e}_y \in \mathbb{R}^d$, similarity and distance measures quantify how closely related the corresponding objects are in the embedding space. In semantic retrieval, these measures are used to rank candidate items according to their geomet-

ric proximity to a query embedding. This follows the vector-space view of semantics, where objects with related meanings are expected to have similar vector representations [21]. In modern sentence and passage retrieval, cosine similarity is commonly used to compare dense embeddings, since it measures similarity in direction rather than raw vector magnitude [23].

One natural way to compare embeddings is the *dot product*, defined as

$$\text{dot}(\mathbf{e}_x, \mathbf{e}_y) = \mathbf{e}_x^\top \mathbf{e}_y.$$

Geometrically, the dot product measures how strongly two vectors point in the same direction, while also depending on their lengths. If two vectors point in similar directions, their dot product is large and positive; if they are nearly orthogonal, it is close to zero; and if they point in opposite directions, it is negative. However, when embeddings are not normalized, the dot product is affected by both semantic direction and vector magnitude. This can be undesirable in retrieval settings where the direction of the vector is intended to encode semantic content.

To remove the effect of vector length, embeddings are often normalized to unit length before comparison. If $\|\mathbf{e}_x\|_2 = 1$ and $\|\mathbf{e}_y\|_2 = 1$, the dot product coincides with cosine similarity. Cosine similarity is defined as

$$\cos(\mathbf{e}_x, \mathbf{e}_y) = \frac{\mathbf{e}_x^\top \mathbf{e}_y}{\|\mathbf{e}_x\|_2 \|\mathbf{e}_y\|_2}.$$

It measures the cosine of the angle between two vectors and takes values in $[-1, 1]$ for nonzero real-valued vectors. A value close to 1 indicates that the vectors point in similar directions, a value close to 0 indicates approximate orthogonality, and a value close to -1 indicates opposite directions. For unit-normalized embeddings, ranking by cosine similarity is therefore equivalent to ranking by dot product.

An alternative way to compare embeddings is through the *Euclidean distance*, which measures the straight-line distance between two points in the embedding space:

$$d_E(\mathbf{e}_x, \mathbf{e}_y) = \|\mathbf{e}_x - \mathbf{e}_y\|_2.$$

Unlike cosine similarity, Euclidean distance treats embeddings as points and measures how far apart they are in absolute space. Smaller Euclidean distance indicates greater proximity. For arbitrary unnormalized vectors, Euclidean distance and cosine similarity can lead to different rankings, because Euclidean distance depends on both direction and magnitude.

For unit-normalized embeddings, however, Euclidean distance and cosine similarity are monotonically related. Expanding the squared Euclidean distance gives

$$\|\mathbf{e}_x - \mathbf{e}_y\|_2^2 = \|\mathbf{e}_x\|_2^2 + \|\mathbf{e}_y\|_2^2 - 2\mathbf{e}_x^\top \mathbf{e}_y.$$

When $\|\mathbf{e}_x\|_2 = \|\mathbf{e}_y\|_2 = 1$, this reduces to

$$\|\mathbf{e}_x - \mathbf{e}_y\|_2^2 = 2 - 2\cos(\mathbf{e}_x, \mathbf{e}_y).$$

Thus, minimizing Euclidean distance is equivalent to maximizing cosine similarity when all embeddings are unit-normalized.

In the multimodal RAG setting studied in this thesis, embeddings produced by the encoder models from Section 2.1.2 are compared using cosine similarity, or equivalently the dot product between normalized embeddings. Retrieved items are then ranked by this similarity score, with higher scores corresponding to items that are closer to the query in the embedding space.

2.2 Large Language Models and the Need for Retrieval

The embedding models introduced in Section 2.1 are used to retrieve relevant evidence, but the final answer in a retrieval-based question answering system is produced by a large language model (LLM). Modern LLMs are commonly based on the transformer architecture, which uses attention mechanisms to model dependencies between tokens in a sequence [27]. Such models can encode substantial knowledge in their parameters, but this parametric knowledge is fixed after training and may be incomplete, outdated, or insufficient for a specific document collection. These limitations motivate the use of external evidence at inference time, rather than requiring all task-specific knowledge to be stored in the model parameters [5].

2.2.1 The Transformer Architecture

The transformer architecture was introduced by Vaswani et al. [27] as a neural-network architecture for sequence modelling based on attention. Instead of relying on recurrence or convolution, a transformer updates token representations using attention mechanisms that allow tokens to exchange information with other tokens in the sequence.

The input text is first tokenized and mapped to a sequence of vector representations. Since the transformer does not process tokens recurrently, it has no built-in notion of word order. Positional information is therefore added to the token representations, allowing the model to distinguish between the same tokens appearing in different positions in the sequence [27].

The central operation in a transformer layer is self-attention. For each token, the model computes attention weights over other tokens in the sequence. These weights determine how much information should be aggregated from each position. The result is a new contextualized representation for each token, where the meaning of a token is influenced by the surrounding sequence. In this way, self-attention provides a mechanism for modelling dependencies between both nearby and distant tokens.

Each transformer layer also contains a position-wise feedforward network, which applies a non-linear transformation to each token representation. The attention mechanism allows tokens to exchange information, while the feedforward network further transforms the resulting representations. By stacking multiple such layers, the model progressively builds increasingly context-dependent token representations [27], [28].

Large language models used for text generation are often decoder-only transformers. In these models, causal masking is applied to self-attention so that each

token can attend only to previous tokens in the sequence. This enables autoregressive generation: the model predicts the next token based on the preceding context, then repeats this process to generate a sequence of text [29], [30]. Formally, for a sequence $\mathbf{x} = (x_1, \dots, x_T)$, an autoregressive language model factorizes the probability of the sequence as

$$p(\mathbf{x}) = \prod_{t=1}^T p(x_t \mid x_{<t}; \theta).$$

This factorization highlights an important property of language models: the output is generated conditionally on the preceding context. The quality and factual grounding of the generated text therefore depend not only on the model parameters, but also on what information is available in the input context.

2.2.2 Parametric Knowledge and Its Limitations

Large language models encode statistical patterns and associations in their parameters θ during training. These parameters allow the model to generate fluent and coherent text, and can be viewed as a form of *parametric memory*. However, this memory is fixed after training. As a result, a model cannot directly access information that was absent from, underrepresented in, or updated after its training data. Purely parametric generation is therefore poorly suited for tasks that require current, specialized, or corpus-specific information [5], [6].

This limitation is especially important in document-based question answering. A query may depend on information contained in a specific document collection, rather than on general knowledge learned during pre-training. If the relevant information is not represented in the model parameters, the model may still produce a plausible answer, but the answer may be unsupported or factually incorrect. This type of fluent but nonfactual generation is commonly referred to as hallucination [3], [4].

The central implication is that an LLM should not be treated as the sole source of knowledge in settings where answers depend on a specific external corpus. Instead, relevant evidence must be made available in the model input context. The next section formalizes how this evidence is retrieved and used in a retrieval-augmented generation pipeline.

2.3 Dense Retrieval and Retrieval-Augmented Generation

With embeddings defined and the limitations of parametric knowledge established, this section assembles the retrieval-augmented generation (RAG) pipeline used in this thesis. The pipeline consists of two stages: candidate retrieval and retrieval-conditioned generation. Together, these stages define the retrieval-generation pattern that later systems can extend to multimodal documents. The overall structure of this pipeline is illustrated in Figure 2.2.

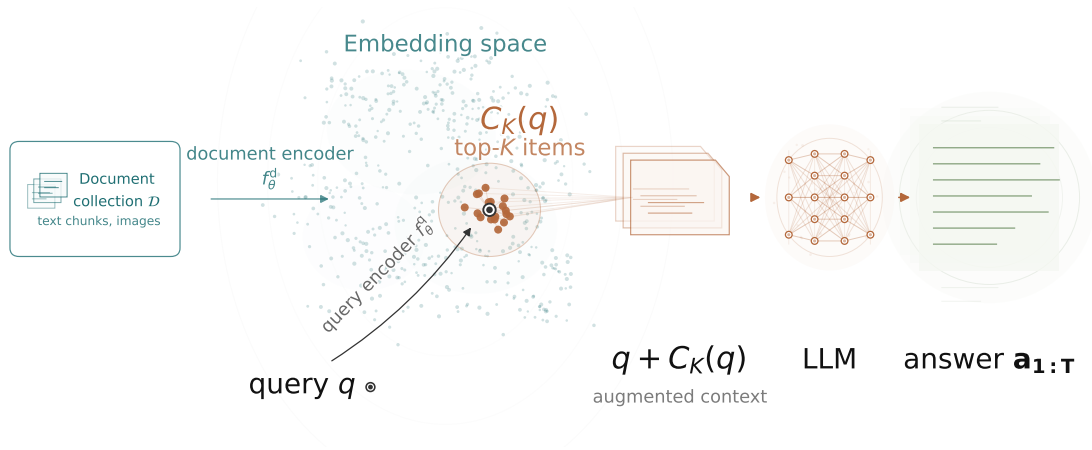


Figure 2.2: Illustrative overview of a retrieval-augmented generation (RAG) pipeline. A query q is used to retrieve a set of relevant items $C_K(q)$ from an embedding space. The retrieved evidence is combined with the query to form an augmented context, which is provided to a language model to generate the answer.

2.3.1 Dense Retrieval and Approximate Indexing

Retrieval can be formulated as a nearest-neighbor search problem in the embedding space introduced in Section 2.1. Given a query q , the goal is to identify the collection items whose embeddings are most similar to the query embedding. In dense retrieval, a so-called bi-encoder is commonly used: one encoder maps queries to vectors and another encoder maps candidate items to vectors, and both outputs live in the same vector space [17], [23], [31]. Let f_{θ}^q denote the query encoder and f_{θ}^d denote the document or item encoder. These functions may be separate, or they may share parameters while using different input formatting for queries and passages. For example, E5-style models are instruction-tuned for retrieval and use role-specific prefixes to distinguish query and passage inputs while still embedding both into the same retrieval space.

For a collection $\mathcal{D}$, the query and each item $d \in \mathcal{D}$ are represented as

$$\mathbf{e}_q = f_{\theta}^q(q), \quad \mathbf{e}_d = f_{\theta}^d(d).$$

Retrieval then corresponds to selecting the top- K items that maximize a similarity function, typically cosine similarity:

$$C_K(q) = \text{top-}K_{d \in \mathcal{D}} \cos(\mathbf{e}_q, \mathbf{e}_d). \quad (2.1)$$

In this thesis, the indexed items are sub-document units extracted from PDF documents, including text chunks and visual elements such as figures, maps, and diagrams.

Dense retrieval differs from sparse lexical retrieval methods such as BM25, which rank documents using term-frequency and document-length statistics derived from query-term overlap [32]. Because dense retrieval compares learned vector representations, it can retrieve items that are semantically related to the query even when there is little or no exact lexical overlap between them. This property is important

in document question answering, where a user query and a relevant passage may express the same information using different words.

A practical challenge is that exact nearest-neighbor search requires comparing the query embedding against every item in the collection, resulting in $\mathcal{O}(|\mathcal{D}|)$ similarity computations per query. This becomes expensive for large corpora. For this reason, dense retrieval systems typically rely on approximate nearest-neighbor (ANN) indexing methods. A widely used method is Hierarchical Navigable Small World (HNSW), a graph-based ANN algorithm that organizes vectors in a multi-layer proximity graph to support efficient approximate search with high recall [33].

2.3.2 Retrieval-Conditioned Generation

Given a query q and a retrieved evidence set $C_K(q)$, the generation stage produces an answer conditioned on both the query and the retrieved evidence. This is the central idea of retrieval-augmented generation: a language model is combined with an external retrieval component, so that generation can depend on information retrieved from a corpus rather than only on knowledge stored in the model parameters [5]. In the terminology of Lewis et al., the language model provides parametric memory, while the retrieved index acts as non-parametric memory.

Formally, let $\mathbf{a} = (a_1, \dots, a_T)$ denote the generated answer sequence. A language model with parameters θ defines a conditional probability distribution over the answer given the query and retrieved evidence:

$$p(\mathbf{a} \mid q, C_K(q); \theta) = \prod_{t=1}^T p(a_t \mid \mathbf{a}_{<t}, q, C_K(q); \theta). \quad (2.2)$$

In practice, the retrieved items are inserted into the model input together with the query, forming an augmented context. The language model then generates an answer conditioned on this context. By grounding generation in retrieved evidence, RAG systems can improve factuality and reduce unsupported generation in knowledge-intensive tasks [5], [6], [8]. The overall performance of such systems is therefore strongly constrained by the retrieval stage: if $C_K(q)$ does not contain the relevant evidence, the generator cannot reliably use that evidence when producing the answer.

2.4 Multimodal and Visual Document Embeddings

Many real-world document collections are inherently multimodal. In the Swedish municipal PDF corpus studied in this thesis, essential information is distributed across both textual and visual elements: maps encode spatial regulations, diagrams convey planning decisions, and figures present data that cannot always be recovered by text extraction alone. This section describes how visual content can be represented as embeddings suitable for retrieval, and identifies the core challenge that arises when text and image embeddings are combined in a joint index.

2.4.1 Visual Information in Documents

Many traditional document retrieval pipelines represent documents primarily as plain text, either through direct extraction for born-digital PDFs or through optical character recognition (OCR) for scanned pages. While this recovers linguistic content, it can discard or weaken information about layout, spatial relationships, graphical marks, figures, tables, fonts, and other visual cues that contribute to document meaning [14], [15], [16]. For document types where meaning is partly encoded visually, such as traffic regulation maps and urban planning diagrams, text-only representations provide only a partial view of document content. Dense retrievers built only on extracted text may therefore fail to surface visually encoded information in response to natural language queries [14], [15].

Recent approaches address this limitation by representing documents at the page or image level rather than relying only on extracted text. ColPali, for example, renders PDF pages as images and uses a vision–language model to produce multi-vector page embeddings, which are compared to queries using a late-interaction retrieval mechanism [14]. The same work introduces ViDoRe, a benchmark for visually rich document retrieval spanning multiple domains, languages, and retrieval settings [14]. These methods show the value of preserving visual document information during retrieval. However, they typically require documents to be re-indexed with a new visual retrieval model, which may be computationally expensive and incompatible with an existing text-based retrieval infrastructure.

2.4.2 Vision–Language Embedding Models

A widely adopted paradigm for embedding images is the bi-encoder setup, in which a vision encoder f_{θ}^v maps images and a text encoder f_{θ}^t maps captions or descriptions into a shared fixed-dimensional vector space. In CLIP-style training, the image encoder and a corresponding text encoder are trained so that matching image–text pairs receive high similarity, while non-matching pairs are pushed apart [11], [25]. CLIP established this paradigm at scale by jointly training image and text encoders to maximize the cosine similarity of matching image–text pairs while minimizing the similarity of incorrect pairings [11]. Subsequent models such as SigLIP refine this approach through a pairwise sigmoid loss for language–image pre-training, avoiding the global softmax normalization used in standard contrastive learning [26]. The resulting embeddings are comparable by cosine similarity within the space in which they were trained, making them suitable for nearest-neighbor retrieval.

Recent multimodal embedding models extend this idea by training a single embedding model to represent several modalities in one shared space. Gemini Embedding 2, for example, maps text, images, video, audio, and documents into a single embedding space for multimodal retrieval and classification [34]. Such models are a natural choice when building a new multimodal retrieval system from scratch. However, they do not remove the practical challenge considered in this thesis: an existing text-based retrieval index may already contain a large corpus of embedded passages. Replacing that index with a new multimodal embedding model would require re-embedding and re-indexing the text collection, whereas a post-hoc alignment approach aims to extend the existing text space with visual content.

When a vision–language encoder is deployed alongside an independently trained text encoder, as is common when extending an existing text retrieval pipeline to support visual content, the two models produce embeddings in geometrically incompatible spaces. Even if both sets of vectors lie on the unit hypersphere and are compared by cosine similarity within their own spaces, their absolute positions and score distributions are not necessarily comparable across spaces.

In document collections where meaning is partly encoded visually, this failure mode can be acute: spatial relationships, graphical symbols, and layout structure that are essential to interpretation may be lost entirely during text extraction, or rendered as uninterpretable fragments. A text-only retrieval index therefore cannot surface this content regardless of how the query is phrased. Resolving the geometric incompatibility between visual and text embedding spaces requires explicit alignment methods, introduced in Section 2.5.

2.5 Embedding Space Alignment

This section establishes the theoretical framework underlying cross-modal retrieval alignment. It first characterizes why independently trained embedding spaces cannot be ranked together directly, then introduces embedding-space translation as a principled approach to bridging them, drawing on two lines of prior work. It then examines why extending this idea to multimodal retrieval is qualitatively more demanding than the text-to-text precedent.

2.5.1 Cross-Modal Incompatibility

Let $\mathbf{e}^t \in \mathbb{R}^{d_t}$ denote an embedding produced by a text encoder and $\mathbf{e}^v \in \mathbb{R}^{d_v}$ an embedding produced by an image encoder. Within each space, normalized vectors can be compared using cosine similarity. However, cosine similarity is meaningful only relative to the geometry of the space in which the embeddings were learned. Different encoders may define spaces with different dimensionalities, orientations, density structures, and score distributions. A similarity score computed within one space carries no reliable interpretation when applied across spaces.

Text and image embeddings also represent structurally different kinds of information. Text embeddings are shaped by linguistic and semantic regularities, while image embeddings are shaped by visual properties such as shape, colour, texture, and spatial layout. Although the two modalities can refer to the same underlying objects, events, or documents, they encode this information along different dimensions and with different granularity. Embeddings from separately trained encoders therefore do not automatically form a shared ranking space, even when their underlying content is related. A similarity score computed in one embedding space has no guaranteed calibration relative to a score computed in another independently trained space.

A related phenomenon arises even when image and text encoders are trained jointly. Liang et al. [35] describe the *modality gap*: in multimodal contrastive models such as CLIP, image and text embeddings tend to occupy separated regions of

the shared representation space. Their analysis attributes this separation to the interaction between model initialization and contrastive optimization dynamics. The modality gap shows that cross-modal comparability is a learned geometric property, not a consequence of normalization alone. It persists even under explicit joint training pressure, which suggests that bridging independently trained spaces requires deliberate alignment beyond what normalization or dimensionality matching can provide.

In joint retrieval, this incompatibility appears not only as a semantic alignment problem but also as a score-compatibility problem. Even if visual items are mapped near relevant text queries, their similarity scores must be on a scale that allows them to compete with text candidates in the same ranked list. If image scores are systematically lower than text scores, visual evidence will be suppressed; if they are systematically higher, images may displace relevant text. Multimodal retrieval therefore requires both semantic alignment and score calibration.

2.5.2 Embedding-Space Translation

Embedding-space translation addresses the incompatibility between spaces by learning a mapping from one space to another, without retraining either encoder. The central question is whether retrieval-useful structure is preserved under such a transformation — that is, whether a mapped embedding can rank its true matches above distractors in the target space, even if it does not reconstruct the target vector exactly.

This was demonstrated concretely by Gao [18], who trained a compact neural network to convert open-source MPNet embeddings into the proprietary `text-ada-002` embedding space. The same texts were embedded with both models, producing paired source and target vectors, and the network was trained to predict the target-space embedding from the source-space embedding. Despite the substantial architectural and dimensional differences between the two models, the trained network achieved an average cosine similarity of 0.932 on held-out data, and the resulting synthetic embeddings could retrieve highly relevant results in a `text-ada-002` index. The key finding is not the specific pair of models used, but the observation that retrieval-useful structure survives a learned transformation between independently trained spaces.

Formally, let $\mathbf{e}_i^s \in \mathbb{R}^{d_s}$ be a source-space embedding and $\mathbf{e}_i^t \in \mathbb{R}^{d_t}$ the corresponding target-space embedding for the same item. A mapping

$$g_\phi : \mathbb{R}^{d_s} \rightarrow \mathbb{R}^{d_t}$$

is trained so that $g_\phi(\mathbf{e}_i^s)$ approximates $\mathbf{e}_i^t$. A general supervised alignment objective can be written as

$$\min_{\phi} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(g_\phi(\mathbf{e}_i^s), \mathbf{e}_i^t), \quad (2.3)$$

where $\mathcal{L}$ measures discrepancy between the mapped source embedding and its target.

A deeper theoretical account of why such mappings can work is offered by Jha et al. [19], who build on the Platonic Representation Hypothesis [36] to propose a *universal geometry* hypothesis for text embeddings: independently trained models

of sufficient capacity tend to converge on a shared latent semantic structure, in the sense that the relative arrangement of semantically similar items is approximately preserved across models even when their absolute coordinates differ. Under this hypothesis, the embedding spaces of different models are not arbitrary coordinate systems but different realizations of a common underlying geometry. Jha et al. show that this structure is strong enough to enable *unsupervised* translation — their Vec2Vec method learns to map between embedding spaces using only unpaired samples from each, without access to the original encoders or to any matched pairs, achieving cosine similarity up to 0.92 to ground-truth target vectors. The implication for alignment is significant: the success of embedding-space translation does not depend entirely on the availability of paired supervision. The spaces themselves carry recoverable geometric correspondence.

2.5.3 Extending Embedding-Space Translation to Multimodal Retrieval

Both the supervised Vec2Vec setting and the unsupervised universal geometry approach rest on a common assumption: the source and target embeddings represent the same underlying object, encoded by two different models. In the supervised case, the same text is passed through two encoders to produce paired training examples. In the unsupervised case, the same texts are embedded by different models and the shared semantic structure is recovered from unpaired samples. Either way, the alignment task is fundamentally a change of coordinates between two representations of the same kind of content.

Extending embedding-space translation to multimodal retrieval breaks this assumption. A text embedding and an image embedding are not two representations of the same input, but two views of potentially related content that may overlap only partially. Text encodes information through linguistic structure and semantic abstraction. Images encode information through visual structure, spatial layout, shape, colour, and texture. Both modalities may refer to the same place, object, or document, but each also contains information that is absent or only implicit in the other. A photograph of a traffic sign encodes spatial geometry, colour, and symbol shape that a textual description may only partially convey. A passage describing a zoning regulation encodes legal and linguistic structure that an image of the map it governs may not make explicit at all.

This partial overlap makes multimodal alignment more demanding than its text-to-text counterpart. The universal geometry hypothesis provides some motivation: if different text models share latent semantic structure, it is plausible that a vision-language model trained on image-text pairs has organized visual content around some of the same semantic axes as a text encoder. But the hypothesis weakens in the multimodal case, since the two modalities do not encode the same information. The alignment objective cannot assume a complete one-to-one correspondence between the two spaces, because the two modalities only partially overlap in the information they encode. Instead, the mapping must identify what information the two modalities share and express that shared structure in the geometry of the target space, while remaining informative despite the content that each modality carries

alone.

For retrieval, the operational goal is that a text query embedding $\mathbf{e}^q$ can be meaningfully compared with a mapped visual embedding $g_\phi(\mathbf{e}^v)$ using the similarity measure of the target space:

$$\cos(\mathbf{e}^q, g_\phi(\mathbf{e}^v)).$$

Achieving this requires two conditions. First, the visual encoder must preserve information about the image that is relevant to the queries the system will receive. Second, the mapping must express that information in a region of the target space where relevant text queries already assign high similarity. The second condition is what makes multimodal alignment qualitatively different from text-to-text translation: the target space is already occupied by text embeddings with their own geometry, and the mapping must place visual embeddings not merely into that space, but into positions that are semantically coherent with respect to the text that is already there.

2.6 Evaluation of Retrieval and VQA-style Tasks

Evaluating a retrieval system requires metrics that measure whether relevant evidence appears among the highest-ranked results for a query. In multimodal retrieval, this must be defined across modalities: a natural language query may need to retrieve a text passage, an image, or another visual document element. Evaluation must therefore specify both what counts as relevant evidence and how success is measured in a ranked list.

This section introduces the retrieval metrics used in the thesis and a VQA-style setup for creating image retrieval queries. In standard visual question answering, the task is to answer a natural-language question about an image. Here, the same general idea is used for retrieval: each question is linked to the item that contains the answer, and retrieval is judged by whether that item appears among the top- K results. This makes it possible to evaluate retrieval on its own and compare text-only and multimodal systems in a consistent way.

2.6.1 Retrieval Metrics

Retrieval systems return a ranked list of items for each query. Evaluation therefore concerns whether relevant items appear near the top of this list, where they are most likely to be used by a user or by a downstream model. A common way to evaluate ranked retrieval is to apply a cutoff K and measure success using only the first K retrieved items. Cutoff-based metrics such as Precision@ K , Recall@ K , and Hits@ K are widely used in information retrieval because they focus evaluation on the highest-ranked results [37].

For a query q , let $L_K(q) = (i_1, \dots, i_K)$ denote the ranked list of the top- K retrieved items, ordered by decreasing cosine similarity between the query embedding and each item embedding. If the query has a single ground-truth item i^* , then Hits@ K measures whether this item appears within the top- K list:

$$\text{Hits@}K(q) = \mathbf{1}[i^* \in L_K(q)], \quad (2.4)$$

where $\mathbf{1}[\cdot]$ denotes the indicator function. The mean Hits@ K over a query set $\mathcal{Q}$ is then

$$\text{Hits@}K = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \text{Hits@}K(q). \quad (2.5)$$

This score is the fraction of queries for which the ground-truth item is retrieved within the first K results.

Hits@ K is closely related to Recall@ K . Recall@ K measures the fraction of all relevant items that appear in the top- K results, and is therefore most natural when several items may be relevant to the same query. When there is exactly one relevant item, Hits@ K and Recall@ K are numerically equivalent. The term Hits@ K makes the single-target evaluation setting explicit.

2.6.2 VQA-style Evaluation

Visual question answering (VQA) is commonly formulated as a task in which a system receives a visual input and a natural language question, and produces a textual answer [38]. Standard VQA evaluation measures answer quality directly, typically by comparing the generated answer against human reference answers. This evaluates the full question-answering process, but it also combines several components: visual understanding, language generation, and answer matching.

For retrieval evaluation, the task can be reformulated so that the question is used as a query and success is measured by whether the evidence needed to answer the question is retrieved. Under this formulation, each instance consists of a question q and a ground-truth evidence item e^* , such as an image, text passage, or document element. A retrieval system returns a ranked list $(e_1, e_2, \dots)$, and retrieval succeeds at cutoff K if

$$e^* \in \{e_1, \dots, e_K\}. \quad (2.6)$$

Performance can then be measured using Hits@ K , as defined in Section 2.6.1.

This retrieval-style formulation is related to knowledge-intensive VQA settings such as OK-VQA, where answering a visual question may require external knowledge beyond what is directly visible in the image [39]. The difference is that the retrieval formulation does not evaluate the final generated answer. Instead, it isolates the evidence selection problem: whether the system can surface the item from which the answer can be derived.

This makes the evaluation reproducible and modality-agnostic. The same criterion can be applied whether the ground-truth evidence is textual, visual, or multimodal. It therefore measures the property most directly tied to retrieval quality: whether a natural language query can retrieve the evidence needed for downstream question answering.

3

Methods

This chapter describes the design and implementation of a multimodal retrieval pipeline that extends a text-only RAG system to support joint retrieval over text chunks and images extracted from Swedish municipal PDF documents. The central contribution is a learned *bridge model* that projects image embeddings from a frozen SigLIP encoder into the embedding space of a frozen E5 text encoder. This enables a single E5 query vector to retrieve both modalities from a shared index. The chapter first describes the document corpus, preprocessing, and vector indexing. It then presents the dataset construction pipeline, baseline systems, bridge architecture, MAPS training procedure, and evaluation protocol.

The training pipeline, named *MAPS* (Multimodal Alignment with Paired Supervision), is described in Section 3.6.

3.1 System Overview

The system extends a standard dense retrieval pipeline to support joint retrieval over text chunks and images drawn from the same PDF corpus. As shown in Figure 3.1, documents are processed along two parallel branches at indexing time: text chunks are encoded with a frozen E5 model [17] and images are encoded with a frozen SigLIP encoder [26] before being projected into E5 space by a bridge model. The text- and image vectors are both stored in a shared Qdrant index, allowing a single E5 query vector to retrieve text and images simultaneously. The bridge model, its training pipeline, and the evaluation framework are described in the sections that follow.

3.2 Document Corpus and Indexing

This section describes the document collection on which the system is built and the steps taken to prepare it for retrieval. The corpus is first introduced in terms of its provenance and content, followed by an account of how raw PDF documents are converted into text chunks and extracted images. These two streams are then encoded and stored in a vector index that serves as the retrieval backend for all experiments in this thesis.

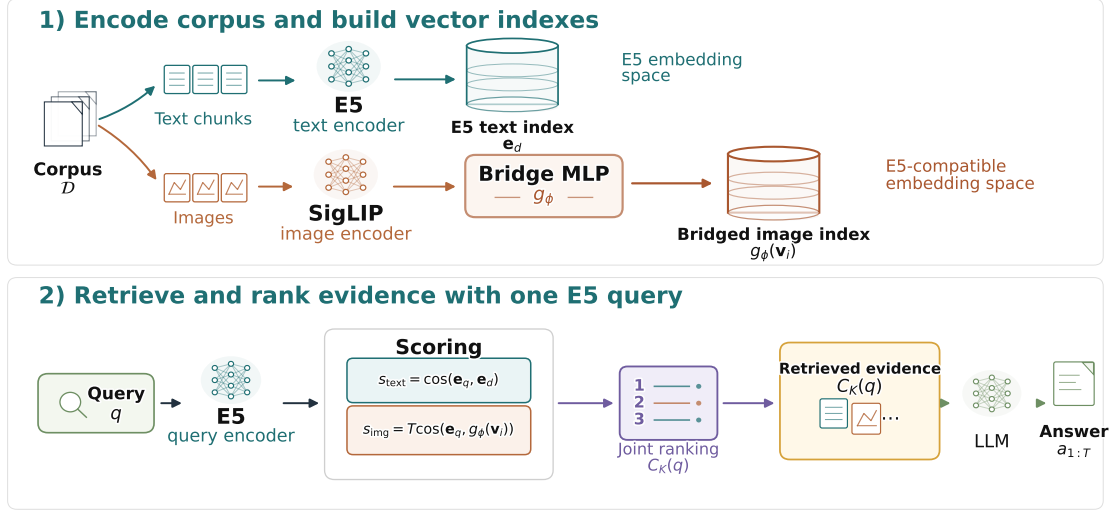


Figure 3.1: Overview of the end-to-end multimodal retrieval pipeline. The document corpus is split into text chunks and images, which are processed along two parallel branches. Text chunks are encoded with E5 and stored in an E5 text index, while images are encoded with SigLIP and mapped by the bridge model g_ϕ into an E5-compatible image index. At query time, a single E5 query embedding scores both indexes, the results are jointly ranked to form the retrieved evidence set $C_K(q)$, and the selected evidence is passed to the LLM to generate the answer $a_{1:T}$.

3.2.1 Corpus

The document collection used in this work is drawn from the SVEA project at AI Sweden [40], a corpus of documents produced by Swedish government agencies and curated by AI Sweden. The corpus covers a wide range of municipal document types, from administrative regulations and planning documents to statistical reports and public information materials. The corpus comprises 235,653 documents and 427,843 extracted images. Images are sparsely distributed across documents, with the large majority of documents containing only one or two images. The extracted images span a range of visual categories, including maps, photographs, charts and diagrams, and logos, reflecting the heterogeneous nature of municipal document collections.

3.2.2 Preprocessing

Each PDF document is converted to markdown prior to indexing using `pdf2md-11m`, an internal preprocessing tool that combines PDF parsing with a vision-language model to extract both text content and image references from each page. Images appear as local file references within the markdown rather than being described in text, and are extracted separately for visual embedding. The markdown is then parsed into chunks using LlamaIndex’s `MarkdownNodeParser`, with `SentenceSplitter` as a fallback for sections that exceed 1024 tokens [41]. The sentence splitter uses an overlap of 100 tokens. Before embedding, image references are stripped from each chunk and whitespace is normalised; each chunk retains metadata linking it to its source file and any image files referenced within it. This yields 901,571 text chunks

and 427,843 images across the full corpus.

3.2.3 Vector Indexing

Three separate vector collections are constructed and stored in a Qdrant vector database [42]. Text chunks are encoded with `intfloat/multilingual-e5-large`, producing an E5 text collection used for all bridge training and evaluation [17]. Text chunks are also encoded with `SigLIP ViT-S0400M`, producing a SigLIP text collection used for the SigLIP multimodal baseline and the bridge training described in Sections 3.4.3 and 3.6, which requires that text and images share the same embedding model. Images are encoded with `SigLIP ViT-S0400M`, producing a SigLIP image collection used in both the baseline and bridge conditions [26]. All vectors are stored alongside metadata linking each vector back to its source document, file path, and any associated image references.

3.3 VQA Dataset Construction

Existing multimodal retrieval benchmarks are predominantly constructed from English-language documents [14], [39], and no multimodal retrieval benchmark covering Swedish municipal documents is known to the author. Beyond language, the visual content of the SVEA corpus is sufficiently domain-specific that general-purpose datasets provide limited coverage: parking maps, planning diagrams, and regulatory photographs require purpose-built evaluation data. Manual annotation at the scale of 427,843 images is infeasible, motivating an automated construction pipeline. The resulting dataset comprises judge-scored image query pairs for multimodal retrieval and judge-scored text query pairs for text retrieval, and is constructed in four stages: image filtering, VQA pair generation, text QA pair generation, and gold filtering.

3.3.1 Image Filtering

Prior to dataset generation, images are filtered to remove content that is unlikely to carry retrievable information. The filtering step assigns each image to one of seven categories using the SigLIP embeddings computed during indexing, avoiding any additional vision-model inference beyond the already computed image embeddings.

For each category c and image embedding $\mathbf{e}_i$, a raw score is computed by averaging cosine similarities against a set of positive prompts $\mathcal{P}_c^+$ and subtracting the average against a set of negative prompts $\mathcal{P}_c^-$:

$$s_{i,c} = \frac{1}{|\mathcal{P}_c^+|} \sum_{\mathbf{p} \in \mathcal{P}_c^+} \cos(\mathbf{e}_i, \mathbf{p}) - \frac{1}{|\mathcal{P}_c^-|} \sum_{\mathbf{p} \in \mathcal{P}_c^-} \cos(\mathbf{e}_i, \mathbf{p}), \quad (3.1)$$

where $\mathbf{p}$ denotes the SigLIP text embedding of a prompt. To account for differences in prompt difficulty across categories, the raw scores are Z-score calibrated column-wise across all N images:

$$\hat{s}_{i,c} = \frac{s_{i,c} - \mu_c}{\sigma_c + \epsilon}, \quad (3.2)$$

where μ_c and σ_c are the mean and standard deviation of $s_{i,c}$ over all images, and ϵ is a small constant for numerical stability. An image is assigned to category $c^* = \arg \max_c \hat{s}_{i,c}$ only if

$$\hat{s}_{i,c^*} \geq z_{\min} \quad \text{and} \quad \hat{s}_{i,c^*} - \hat{s}_{i,c^{(2)}} \geq \delta, \quad (3.3)$$

where $c^{(2)}$ is the second-best category, $z_{\min} = -1$, and $\delta = 0.25$. Images that fail either conditions are left unassigned. The bottom 25 percent of each category, ranked by $\hat{s}_{i,c^*}$, is further discarded to remove low-confidence assignments. Since the scoring relies on visual similarity rather than semantic understanding, visually ambiguous images may be misrouted or left unassigned. The filtering is intended to produce a cleaner subset for generation by excluding logos, stamps, dense text scans, and other low-information visual content. The final filtering scheme therefore uses four empirically defined excluded categories and three generation categories to separate potentially useful visual content from low-information images before VQA generation.

The seven routing categories and their disposition in the pipeline are listed in Table 3.1. The full positive and negative prompt lists for each category are given in Appendix B.

Table 3.1: Image routing categories and their disposition in the dataset construction pipeline.

Category	Disposition	Description
maps	Passed to generation	Aerial maps, street plans, cadastral drawings
photos	Passed to generation	Ground-level photographs of outdoor scenes
charts_diagrams	Passed to generation	Bar charts, flowcharts, schematics, tables
logos_crests	Excluded	Heraldic shields, crests, coat of arms
logos_wordmarks	Excluded	Typography logos, wordmarks
junk_text	Excluded	Text scans, dense legal document pages
junk_stamps	Excluded	Ink stamps, digital signatures

3.3.1.1 Category Label Noise Estimation

Since the routing classifier is prompt-based and heuristic, some images may be assigned to the wrong category. This label noise is estimated empirically by manually inspecting a random sample of 50 images per category drawn from the gold VQA records. Each sampled image is inspected and marked as correctly routed or misrouted based on whether its visual content matches the assigned category. The per-category noise rate is estimated as:

$$\hat{\rho}_c = \frac{|\{i \in \mathcal{S}_c : \text{misrouted}(i)\}|}{|\mathcal{S}_c|}, \quad (3.4)$$

where $\mathcal{S}_c$ is the sample for category c . Automated noise estimation using VQA question embeddings was also explored, but was found unreliable for this corpus. Because the questions were generated under a blind-user constraint, they contain little category-identifying language, resulting in weak separation between categories in E5 embedding space.

3.3.2 VQA Triplet Generation

Each image passes through a three-stage pipeline: a gatekeeper that determines whether the image is useful and how much document context the generator needs, a generator that produces question-answer-explanation triplets, and a judge that verifies factual grounding. Figure 3.2 gives an overview of the full generation pipeline.

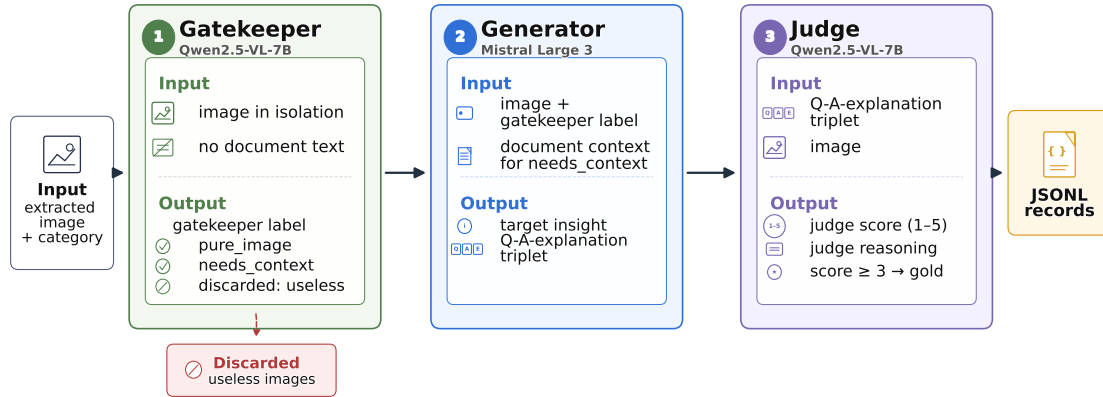


Figure 3.2: The three-stage VQA generation pipeline. Each image is first assessed by a gatekeeper, then passed to a generator that produces a question-answer-explanation triplet, and finally evaluated by a judge that scores factual grounding.

3.3.2.1 Gatekeeper

Each image is first assessed by **Qwen2.5-VL-7B-Instruct**, a vision-language model chosen for its document, chart, diagram, and layout understanding capabilities [43]. The same model serves as the judge in Section 3.3.2.3; given the 16 GB VRAM of the T4, running a second vision-language model in parallel was not feasible. Both stages load the model in 4-bit quantization for memory efficiency. The gatekeeper evaluates each image in isolation, without access to any surrounding document text, and assigns one of three labels. Images classified as **useless**, including logos, UI elements, and decorative photographs that carry no retrievable information, are discarded immediately. Images classified as **pure_image** are deemed self-explanatory and passed to the generator without document context. Images classified as **needs_context** contain data that requires surrounding text to be interpreted correctly, such as a map with unlabelled markings or a chart without a legend, and are passed to the generator together with the four paragraphs preceding and following the image reference in the source markdown file. The full gatekeeper prompts are given in appendix A.1.

3.3.2.2 Generator

Question-answer-explanation triplets are generated by **Mistral Large 3**, a multi-modal mixture-of-experts model with 675 billion total parameters and 41 billion active parameters [44]. The model is hosted on AI Sweden’s infrastructure. Access to AI Sweden’s servers motivates the choice of Mistral over external commercial APIs. All generation is performed at temperature 0.0 with a prompt-based JSON output format to ensure consistent and parseable output.

Rather than generating a question-answer pair alone, the generator is prompted to produce a question, answer, and explanation for each image. This design is inspired by chain-of-thought prompting, where generating an explicit rationale can improve model performance on reasoning tasks [45]. In this pipeline, the explanation is not used as hidden reasoning or as an independent truth source; instead, it provides the judge with an explicit rationale to inspect alongside the proposed answer.

Generation follows one of two modes determined by the gatekeeper label. For **pure_image** inputs, the generator receives only the image. For **needs_context** inputs, the generator receives the image together with the four paragraphs preceding and following the image reference in the source markdown file. In both modes the answer must be physically readable from the image pixels.

Each successful generation produces a structured record containing a target insight, a question, an answer, and an explanation. The target insight is a short description of the specific data point the question focuses on, and is generated before the question itself to anchor the generator to a concrete and retrievable piece of information. Records that cannot be parsed or that produce empty fields are discarded before being passed to the judge.

The generator is instructed to produce a self-contained natural language query that reads as a search a user would perform without knowledge of the image. A blind user constraint is enforced: the question must not reference the image, the map, the document, or any visual element by name. For **needs_context** inputs, the premise of the question may be grounded in the document context, but the answer must be unambiguously readable from the image pixels and not inferable from the text alone. The full generation prompts are provided in Appendix A.2.

3.3.2.3 Judge

Each generated triplet is evaluated by **Qwen2.5-VL-7B-Instruct**, the same model used in the gatekeeper stage [43]. The judge receives the image, the question, the answer, and the explanation, and for **needs_context** inputs it additionally receives the document context that was provided to the generator. This ensures that the judge can verify whether the question premise is legitimately grounded in the context rather than hallucinated.

The judge scores each triplet on a scale of 1 to 5 based on factual grounding, verifying that the answer is physically visible in the image pixels and not inferred or invented. For **needs_context** inputs, the judge is instructed to accept the document context as a given premise and focus verification on whether the answer can be read from the image. The scoring rubric is given in Table 3.2.

Table 3.2: Judge scoring rubric for factual grounding of generated triplets.

Score	Description
1	Serious hallucination – answer is not present in the image
2	Major errors – answer is largely inconsistent with image content
3	Partially grounded – answer is related but contains factual errors
4	Well grounded – answer is correct with only minor issues
5	Perfectly grounded – answer is unambiguously readable from image pixels

The score and the judge reasoning are stored in the output record alongside the generated triplet. Pairs scoring 3 or above are considered gold candidates and pass to the gold filtering stage described in Section 3.3.4. The full judge prompt is provided in Appendix A.3.

3.3.3 Text QA Generation

Text query pairs are generated from the indexed text chunks for use in both training and evaluation of text retrieval. The pipeline follows the same generator and judge stages described in Section 3.3.2, but operates entirely on text without any image input and omits the gatekeeper stage. Instead, a chunk filtering step is applied before generation to ensure that only chunks likely to yield specific, answerable questions are passed to the generator.

3.3.3.1 Generator

Candidate question-answer pairs are generated by **Mistral Large 3** using the same configuration as in Section 3.3.2.2 [44]. Given a text chunk, the generator produces a question, an answer, and a chunk excerpt, the exact verbatim phrase from the chunk that contains the answer. The chunk excerpt serves as a grounding anchor for the judge, providing a precise location in the source text where the answer can be verified. The same blind user constraint applies: the question must read as a natural search query and must not reference the text, the document, or the passage. Questions must target specific factual content such as names, numbers, addresses, dates, or rules rather than vague summaries. The full generation prompt is provided in Appendix A.4.

3.3.3.2 Judge

Each candidate pair is evaluated by **Qwen2.5-VL-7B-Instruct** using the same scoring rubric as in Section 3.3.2.3 (Table 3.2), but operating on text only without any image input [43]. The judge receives the chunk text, the question, the answer, and the chunk excerpt, and verifies whether the answer can be derived directly from the chunk without external knowledge. The presence of the chunk excerpt in the source text is checked explicitly, and the judge is informed whether it was found verbatim, providing an additional grounding signal. Pairs scoring 3 or above are considered

gold candidates and proceed to the gold filtering stage described in Section 3.3.4. The full judge prompt is provided in Appendix A.5.

3.3.4 Gold Filtering

A set of gold-filtering criteria is applied in addition to the judge score threshold, to catch cases where the numerical score alone is insufficient to detect low-quality pairs. A record is retained only if it satisfies all of the following criteria:

- (i) Judge score is 3 or above.
- (ii) Judge reasoning contains no hallucination markers from a predefined keyword list.
- (iii) Error message field is empty.
- (iv) All required fields are present: question, answer, judge reasoning, and image path (VQA) or chunk text (text QA).
- (v) For `pure_image` records: document context must be absent.
- (vi) For `needs_context` records: document context must be present.

Criteria (v) and (vi) are specific to image records and do not apply to text QA.

For the VQA outputs, these criteria are applied as a post-generation filtering step that produces the final gold image dataset, and VQA records failing any criterion are written to a separate rejection file for inspection rather than silently discarded. For the text QA outputs, the applicable criteria, (i)–(iv), are instead enforced inline during generation: candidate pairs that fail any criterion are discarded at generation time and not tracked, so no text QA records are removed at this stage and no separate generated count is available for text QA.

The filtering step produces two separate gold datasets: one comprising judge-scored image query triplets for multimodal retrieval, and one comprising judge-scored text query pairs for text retrieval. The final counts for both datasets are reported in Section 4.1.1.

Both datasets are partitioned into training and held-out evaluation splits prior to any model training. The split sizes and construction procedure are reported in Section 4.1.1.

3.4 Baseline Systems

Three baseline conditions are defined before introducing the bridge, each serving a different diagnostic role. The first evaluates image-only retrieval using a purpose-built language–image encoder. The second evaluates naive joint retrieval, where scores from independently trained text and image embedding spaces are merged without alignment, and serves as a diagnostic condition for score compatibility. The third evaluates a shared SigLIP embedding setup for multimodal retrieval, providing a purpose-built cross-modal baseline against which the bridge can be compared.

3.4.1 SigLIP Image-Only

The first condition evaluates image-only retrieval using SigLIP [26]. A natural language query is encoded with the SigLIP text encoder and ranked against the SigLIP image vectors by cosine similarity. Text chunks are not included in the candidate pool. This condition serves as a diagnostic baseline: it measures image retrieval performance when both the query and image candidates are represented within the same purpose-built language-image embedding space. It also provides a reference point for the bridge image-only condition described in Section 3.7.1.

3.4.2 Naive Joint Retrieval

The second condition evaluates naive joint retrieval without alignment. An E5 query vector is used to score all 901,571 E5 text chunk vectors and a SigLIP query vector to score all 427,843 SigLIP image vectors. The resulting scores are merged into a single ranked list. This condition tests whether cosine scores from the two independently trained embedding spaces can be compared directly. Since the E5 and SigLIP spaces have different geometries and score distributions, this condition is included as a diagnostic baseline for the score compatibility problem that motivates the bridge.

3.4.3 SigLIP Multimodal Baseline

The third condition evaluates a shared SigLIP embedding setup for multimodal retrieval [26]. A SigLIP text query vector is used to score both the SigLIP text chunk vectors and the SigLIP image vectors by cosine similarity, and the resulting scores are merged into a single ranked list. Because SigLIP is trained as a language-image model with separate text and image encoders whose representations are comparable within the same pretraining space, this condition does not require a post-hoc E5-SigLIP bridge [26]. It therefore serves as the primary purpose-built cross-modal baseline for the bridge joint retrieval condition described in Section 3.7.1.

3.5 Bridge Model

The bridge model is a learned mapping from SigLIP image embedding space (1152-dimensional) into E5 text embedding space (1024-dimensional). Because SigLIP and E5 were trained independently with different objectives and on different data, their output vectors are geometrically incompatible: a raw SigLIP image vector cannot be meaningfully compared to an E5 text vector using cosine similarity. The bridge addresses this by learning a projection that places image vectors into the same geometric space as E5 text vectors. Both the SigLIP and E5 encoders are frozen throughout training, meaning the bridge is the only component with trainable parameters and all cross-modal alignment capacity of the system resides in it. At indexing time, SigLIP image vectors are projected through the bridge into E5 space and stored alongside E5 text vectors. At query time, a single E5 query vector retrieves over the resulting joint index.

3.5.1 Architecture

The bridge is a residual MLP consisting of an input projection followed by a stack of BridgeBlocks and a final output normalization, as illustrated in Figure 3.3. The input projection is a single linear layer that maps the 1152-dimensional SigLIP input into the 1024-dimensional E5 output space. Each BridgeBlock applies a pre-norm residual transformation: the input is first passed through a LayerNorm, then through a two-layer feedforward network with a GELU activation and dropout, and the result is added back to the block input via a residual connection. The output of the final block is passed through a LayerNorm before being L2-normalized, placing it on the unit hypersphere and making it directly comparable to E5 vectors under cosine similarity.

The number of BridgeBlocks n controls the capacity of the bridge and is varied in the ablation study described in Section 3.7.3. The default configuration uses $n = 4$ blocks with a dropout rate of 0.1, giving approximately 18 million trainable parameters in total (see Table 4.10 for exact counts at other capacities).

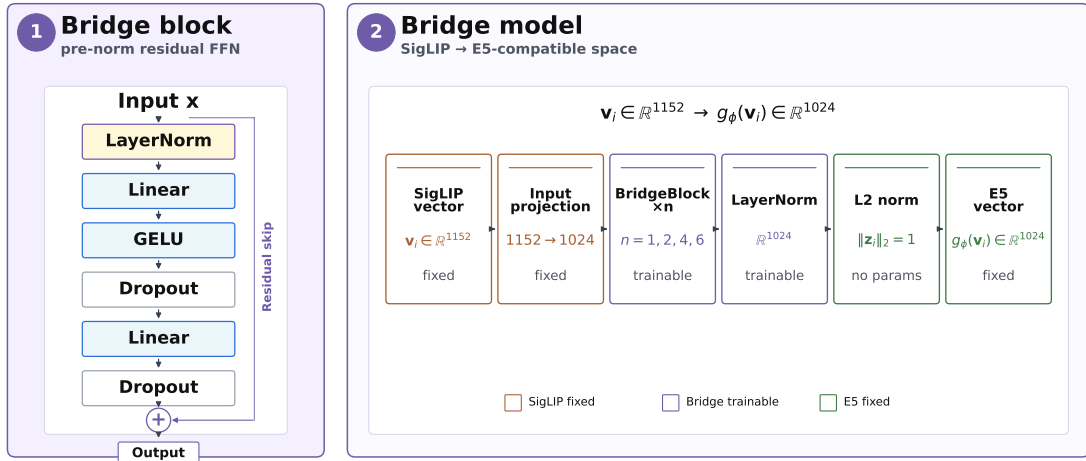


Figure 3.3: Architecture of the bridge model used to project SigLIP image embeddings into the E5 text embedding space. Panel (1) shows the internal structure of a residual BridgeBlock, where the input is processed by a pre-normalized feed-forward network and added back through a residual skip connection. Panel (2) shows the full bridge model: a SigLIP image vector $\mathbf{v}_i \in \mathbb{R}^{1152}$ is mapped to 1024 dimensions by an input projection, passed through a stack of BridgeBlocks, and L2-normalized to produce an E5-compatible representation $g_\phi(\mathbf{v}_i) \in \mathbb{R}^{1024}$ for joint image–text retrieval.

3.6 MAPS Training Pipeline

To train the bridge model, a staged curriculum named *MAPS* (Multimodal Alignment with Paired Supervision) is introduced, designed to move gradually from strong geometric supervision toward the final joint image–text retrieval setting.

A central design assumption of MAPS is that the source encoder is a vision–language model rather than a pure image encoder. SigLIP is trained to embed both images and text in a shared representation space using paired image–text supervision [26]. This means that the same encoder that produces image embeddings for the corpus can also produce text embeddings for the same corpus chunks. Because both the SigLIP text embeddings and the E5 text embeddings of each chunk represent the same underlying text, they form a large-scale set of paired vectors across the two spaces. This satisfies the core assumption of the Vec2Vec framework [18]: that paired embeddings of the same objects are available in both the source and target spaces. The cross-modal alignment problem is therefore not approached from scratch. Instead, the text–text channel provides a direct supervision signal that can be exploited to establish a coarse geometric mapping from SigLIP space into E5 space before any image-specific supervision is introduced.

MAPS exploits this structure through a staged curriculum designed to move gradually from strong geometric supervision toward the final joint retrieval setting. The central design principle is that paired supervision must anchor the bridge semantically before weaker, noisier, or ranking-based signals are introduced. This ordering reflects the weak co-occurrence structure of the corpus, where the median document contains only one or two images and text chunks describe the document as a whole rather than any specific figure. Under these conditions, distribution-matching objectives alone lack the semantic grounding needed to establish reliable cross-modal correspondence. Weaker and noisier stages are therefore reserved for corpus extension, regularisation, and joint ranking rather than initial alignment.

MAPS consists of five sequential stages: target-space shaping, text-to-text pretraining, image-query anchoring, corpus extension, and unified multimodal ranking. Full hyperparameter settings for all stages are listed in Appendix C.

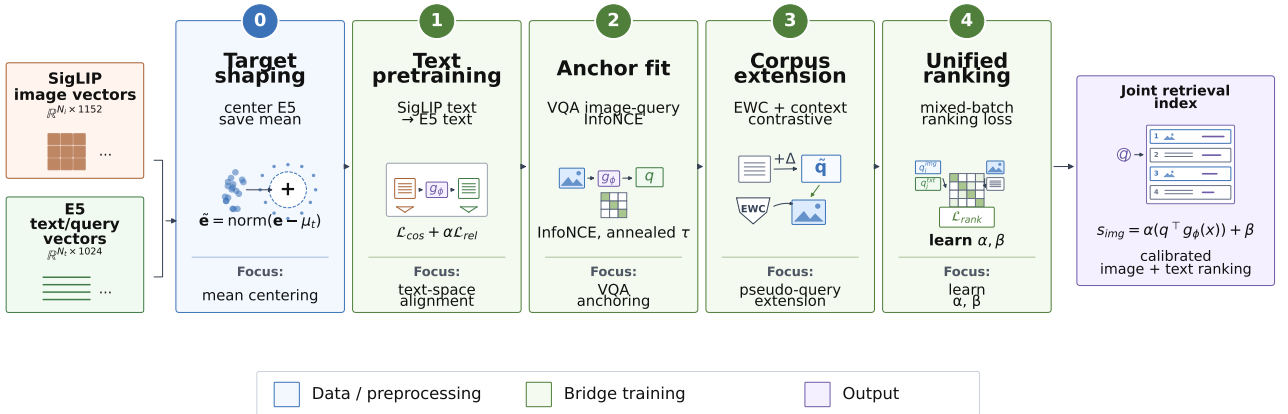


Figure 3.4: MAPS training pipeline. SigLIP image vectors and E5 text/query vectors are brought into a shared retrieval setting through five stages: E5 target-space shaping, text-to-text bridge pretraining, VQA-based image–query anchoring, corpus-level pseudo-query extension, and unified image–text ranking. The final stage learns the image score calibration parameters used when projected images and E5 text chunks compete in the same joint retrieval index.

3.6.1 Stage 0: Target Space Shaping

Before bridge training begins, the E5 target space is preprocessed to improve the quality of cosine-based similarity comparisons. Raw E5 embeddings are highly anisotropic: most vectors share a strong common direction, which compresses the effective range of cosine similarities and weakens discriminative power in contrastive training [20], [46]. Stage 0 removes this bias by estimating and subtracting the empirical mean of the E5 corpus distribution.

Let $\{\mathbf{e}_j^t\}_{j=1}^M$ denote the full set of E5 text embeddings from the corpus. The empirical mean is computed as

$$\boldsymbol{\mu} = \frac{1}{M} \sum_{j=1}^M \mathbf{e}_j^t. \quad (3.5)$$

Each E5 vector is then centered by subtracting $\boldsymbol{\mu}$ and renormalized to unit length:

$$\tilde{\mathbf{e}}_j^t = \frac{\mathbf{e}_j^t - \boldsymbol{\mu}}{\|\mathbf{e}_j^t - \boldsymbol{\mu}\|_2}. \quad (3.6)$$

The mean vector $\boldsymbol{\mu}$ is saved and applied consistently throughout all subsequent stages: E5 query vectors and E5 chunk vectors are centered in the same way before being compared against bridge outputs. At inference time, the same centering is applied to queries before scoring against the bridge-projected image index.

3.6.2 Stage 1: Text-to-Text Pretraining

Stage 1 warm-starts the bridge using large-scale paired text embeddings before any image supervision is introduced. For every text chunk in the corpus, both a SigLIP text embedding and an E5 text embedding are available, since the same chunks were indexed with both encoders during corpus construction (Section 3.2.3). Because both embeddings represent the same underlying text, this provides a natural source of paired supervision at corpus scale without requiring any image–query annotations.

The bridge is trained to map SigLIP text vectors into the centered E5 text space:

$$g_\phi(\mathbf{e}_i^{s,\text{text}}) \approx \tilde{\mathbf{e}}_i^{t,\text{text}}, \quad (3.7)$$

where $\mathbf{e}_i^{s,\text{text}}$ is the SigLIP text embedding and $\tilde{\mathbf{e}}_i^{t,\text{text}}$ is the centered and renormalized E5 text embedding from Stage 0. The training objective combines a cosine alignment term with a relational structure term:

$$\mathcal{L}_{\text{text}} = \mathcal{L}_{\text{cos}} + \alpha \mathcal{L}_{\text{rel}}, \quad (3.8)$$

where $\mathcal{L}_{\text{cos}}$ minimizes the cosine distance between each projected SigLIP text vector and its corresponding centered E5 vector,

$$\mathcal{L}_{\text{cos}} = \frac{1}{B} \sum_{i=1}^B \left(1 - g_\phi(\mathbf{e}_i^{s,\text{text}})^\top \tilde{\mathbf{e}}_i^{t,\text{text}} \right), \quad (3.9)$$

and $\mathcal{L}_{\text{rel}}$ preserves the pairwise relational structure of the target space by penalizing differences between the off-diagonal similarity matrices of the projected and target vectors:

$$\mathcal{L}_{\text{rel}} = \frac{1}{B(B-1)} \sum_{i \neq j} \left(g_{\phi}(\mathbf{e}_i^{s,\text{text}})^{\top} g_{\phi}(\mathbf{e}_j^{s,\text{text}}) - \tilde{\mathbf{e}}_i^{t,\text{text}}{}^{\top} \tilde{\mathbf{e}}_j^{t,\text{text}} \right)^2, \quad (3.10)$$

where B is the number of pairs in the training batch. The relational term encourages the bridge to reproduce not only individual vector directions but also the neighbourhood geometry of the E5 space. This is important because Stage 2 will rely on that geometry to pull image vectors toward semantically relevant query regions.

The motivation for placing this stage before image supervision is that the SigLIP and E5 spaces share a modality-neutral text channel: the same sentence can be embedded by both encoders. Exploiting this channel establishes a coarse geometric mapping from SigLIP space to E5 space before the harder cross-modal image alignment is introduced in Stage 2.

3.6.3 Stage 2: Anchor Fit

Stage 2 introduces the image–query pairs from the VQA dataset and uses them to anchor the bridge in the cross-modal direction. For each training example, the SigLIP image embedding $\mathbf{e}_i^{s,\text{img}}$ is passed through the bridge and compared against the centered E5 embedding of the corresponding natural language question $\tilde{\mathbf{q}}_i^t$. The training objective is an InfoNCE loss with cosine-annealed temperature:

$$\mathcal{L}_{\text{anchor}} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(g_{\phi}(\mathbf{e}_i^{s,\text{img}})^{\top} \tilde{\mathbf{q}}_i^t / \tau)}{\sum_{j=1}^B \exp(g_{\phi}(\mathbf{e}_i^{s,\text{img}})^{\top} \tilde{\mathbf{q}}_j^t / \tau)}, \quad (3.11)$$

where in-batch negatives are used: for each image i , all other questions in the batch serve as negatives. The temperature τ is annealed from a warm starting value of 0.2 down to 0.07 over the course of training. The warm start prevents vanishing gradients early in training, when the bridge is still poorly initialized and all logits are near-uniform at low temperature. Annealing toward the temperature of 0.07, used in the CLIP paper [11], sharpens the contrastive objective as alignment improves.

Stage 2 provides the strongest and most direct cross-modal supervision available in the pipeline. Each image is paired with a verified natural language question that was generated specifically from that image and validated by the LLM judge described in Section 3.3.2.3. This makes the image–query correspondence reliable rather than approximate. The geometric mapping established by Stage 1 in the SigLIP–E5 text direction is now refined to pull image vectors toward the query regions of the centered E5 space that are relevant to their visual content.

3.6.4 Stage 3: Corpus Extension

Stage 2 anchors the bridge using the verified VQA pairs, but the full corpus contains 427,843 images. The majority of these images have no paired query annotation and are never seen during Stage 2 training. Stage 3 extends the alignment to the

full corpus image collection using weaker supervision signals derived from document structure. It proceeds in two sequential phases: an EWC anchor phase that consolidates the Stage 2 alignment before noisy corpus signals are introduced, followed by a context-grounded contrastive phase that uses surrounding document text to supervise the unannotated images.

3.6.4.1 Phase 1: EWC Anchor

Before corpus-scale supervision is introduced, the bridge parameters are stabilized with respect to the VQA-aligned directions learned in Stage 2. Naively fine-tuning on corpus images risks catastrophic forgetting: gradient updates from the new objective can overwrite the cross-modal alignment that Stage 2 established. To prevent this, Phase 1 applies Elastic Weight Consolidation (EWC) [47], which penalizes deviations from the Stage 2 parameters weighted by their importance to the VQA alignment task.

The importance of each parameter θ_i is estimated by its Fisher information, approximated from a sample of N_F VQA pairs:

$$F_i \approx \frac{1}{N_F} \sum_{n=1}^{N_F} \left(\frac{\partial \mathcal{L}_{\text{anchor}}}{\partial \theta_i} \right)^2, \quad (3.12)$$

where $\mathcal{L}_{\text{anchor}}$ is the Stage 2 InfoNCE loss evaluated on each VQA sample. The EWC penalty then discourages changes to parameters that were important for VQA alignment:

$$\mathcal{L}_{\text{EWC}} = \frac{\lambda}{2} \sum_i F_i (\theta_i - \theta_i^*)^2, \quad (3.13)$$

where θ_i^* denotes the Stage 2 parameter values and λ controls the strength of the penalty. The combined Phase 1 objective is:

$$\mathcal{L}_1 = \mathcal{L}_{\text{consistency}} + \mathcal{L}_{\text{EWC}}, \quad (3.14)$$

where $\mathcal{L}_{\text{consistency}}$ is an MSE loss between the bridge output on augmented corpus images and the outputs of the Stage 2 checkpoint loaded as a frozen reference model. Unlike a momentum teacher, this reference model’s weights are not updated during training; it serves purely as a fixed target to prevent the bridge from drifting far from its Stage 2 behaviour on unannotated corpus images. The EWC penalty simultaneously protects the VQA-specific alignment by penalizing changes to the parameters that were most important for verified image–query correspondence.

3.6.4.2 Phase 2: Context-Grounded Contrastive Learning

The second phase supervises the full corpus image collection using pseudo-queries derived from surrounding document text. The motivation is that visually similar images in this corpus can have different semantic content: two cadastral maps from different municipal decisions may share layout, symbols, and visual style while referring to different geographic areas. The SigLIP encoder may not distinguish them reliably by visual appearance alone. Document text, however, is specific to each

image’s context and can provide the semantic signal needed to separate them in E5 space.

This supervision signal is weak and noisy. Document chunks describe the source document as a whole rather than the specific image they surround, and the passage-to-query offset is estimated from a limited set of VQA pairs. The resulting pseudo-queries are therefore approximate and may point in incorrect directions for some corpus images. Three design choices work together to reduce the risk of this noise degrading the Stage 2 alignment. First, a confidence gate filters out pseudo-labels that are too distant from real VQA queries, removing the lowest-quality supervision before training begins. Second, a soft contrastive temperature dampens the gradient contribution of any single pseudo-label relative to the batch average, reducing the risk of large destructive updates. Third, the EWC penalty from Phase 1 is carried forward: it constrains the bridge to stay close to its Stage 2 parameters along the directions that mattered most for verified image–query alignment, allowing the weak corpus signal to be absorbed without overwriting the alignment established from verified pairs.

3.6.4.2.1 Pseudo-label construction. For each corpus image, the four paragraphs preceding and following its reference in the source markdown file are extracted as document context. This context is embedded with E5 to produce a passage vector $\mathbf{e}_{\text{ctx}}$. Because E5 passage vectors and E5 query vectors may occupy different regions of the embedding space, a passage-to-query offset Δ is estimated from the VQA training pairs:

$$\Delta = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} (\tilde{\mathbf{q}}_i^t - \mathbf{e}_{i,\text{ctx}}), \quad (3.15)$$

where $\mathcal{S}$ is the set of `needs_context` VQA pairs and $\mathbf{e}_{i,\text{ctx}}$ is the centered E5 embedding of the surrounding context for image i . The pseudo-query for each corpus image is then constructed as:

$$\tilde{\mathbf{q}}_i^{\text{pseudo}} = \frac{\mathbf{e}_{i,\text{ctx}} + \Delta}{\|\mathbf{e}_{i,\text{ctx}} + \Delta\|_2}. \quad (3.16)$$

A confidence gate then retains only images whose pseudo-query exceeds a minimum cosine similarity τ_{gate} to its nearest neighbor in the set of `needs_context` VQA query vectors. Using only `needs_context` queries as the gate reference ensures that pseudo-labels are accepted only for images that genuinely benefit from document context, leaving the `pure_image` alignment from Stage 2 less affected by the corpus-extension stage.

3.6.4.2.2 Three-term loss. Phase 2 optimizes a three-term objective that combines the contrastive pseudo-label signal, a geometric regulariser, and the EWC anchor carried forward from Phase 1:

$$\mathcal{L}_2 = \mathcal{L}_{\text{InfoNCE}} + \lambda_1 \mathcal{L}_{\text{Lap}} + \lambda_2 \mathcal{L}_{\text{EWC}}. \quad (3.17)$$

The InfoNCE term uses pseudo-queries as positives with a soft temperature $\tau = 0.2$ to account for label noise:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(g_\phi(\mathbf{e}_i^{s,\text{img}})^\top \tilde{\mathbf{q}}_i^{\text{pseudo}} / \tau)}{\sum_{j=1}^B \exp(g_\phi(\mathbf{e}_i^{s,\text{img}})^\top \tilde{\mathbf{q}}_j^{\text{pseudo}} / \tau)}. \quad (3.18)$$

The Laplacian regulariser $\mathcal{L}_{\text{Lap}}$ provides a geometric constraint on the bridge output. It enforces manifold smoothness: images that are visually similar in SigLIP space should map to nearby points in E5 space. This is particularly important under noisy pseudo-label supervision, where the contrastive objective alone may push visually related images apart if their pseudo-queries differ. Let $\mathbf{F} \in \mathbb{R}^{B \times d}$ denote the matrix of bridge outputs for a batch, and let $\mathbf{L} = \mathbf{D} - \mathbf{A}$ be the graph Laplacian constructed within the batch, where the edge weight between images i and j is:

$$w_{ij} = \max(0, \cos(\mathbf{e}_i^{s,\text{img}}, \mathbf{e}_j^{s,\text{img}})), \quad (3.19)$$

and $\mathbf{D}$ is the diagonal degree matrix with $D_{ii} = \sum_j w_{ij}$. The regulariser is the trace of the Laplacian quadratic form:

$$\mathcal{L}_{\text{Lap}} = \frac{1}{B^2} \text{tr}(\mathbf{F}^\top \mathbf{L} \mathbf{F}) = \frac{1}{B^2} \sum_{i,j} w_{ij} \|g_\phi(\mathbf{e}_i^{s,\text{img}}) - g_\phi(\mathbf{e}_j^{s,\text{img}})\|^2. \quad (3.20)$$

This penalizes the bridge for mapping visually similar images to distant points in E5 space. The graph is constructed within each batch from SigLIP cosine similarities, requiring no full-corpus graph at training or inference time. The InfoNCE term prevents the trivial collapse solution in which all outputs are identical, since identical outputs would yield a uniform distribution over negatives and a high contrastive loss. The EWC term $\mathcal{L}_{\text{EWC}}$ is identical to Equation (3.13) and is carried forward from Phase 1 to prevent the context-grounded objective from overwriting the verified alignment established in Stage 2.

3.6.5 Stage 4: Unified Multimodal Ranking

Stages 1–3 train the bridge to place image vectors in the correct region of the E5 space, but they do not train it in the setting where images and text actually compete. At inference time, a single query must rank correctly against a pool containing both projected images and E5 text chunks. Stage 4 therefore introduces a mixed image–text ranking objective directly. Each training batch contains both image–query pairs and text–query pairs, and the model is optimized on a single candidate pool in which projected images and text chunks compete under the same cross-entropy ranking loss.

In addition to the bridge parameters, Stage 4 learns two image-score calibration parameters: a multiplicative scale and an additive bias. These parameters are applied only to image scores, while text scores remain unchanged. This allows projected image scores to become competitive with text scores without changing the geometry of the text embedding space.

3.6.5.1 Problem Formulation

Each training batch samples $B/2$ image–query pairs from the VQA dataset and $B/2$ text–query pairs from the text QA dataset. Let $\mathbf{Z}^{\text{img}} = g_\phi(\mathbf{E}^{s,\text{img}})$ denote the projected image vectors, and let $\tilde{\mathbf{E}}^{t,\text{text}}$ denote the centered E5 text chunk vectors. The query matrix is formed by concatenating the centered E5 query vectors for image and text examples:

$$\mathbf{Q} = [\tilde{\mathbf{Q}}^{\text{img}} ; \tilde{\mathbf{Q}}^{\text{txt}}] \in \mathbb{R}^{B \times d}. \quad (3.21)$$

The candidate matrix contains the projected images and the text chunks, stacked in the same order:

$$\mathbf{C} = [\mathbf{Z}^{\text{img}} ; \tilde{\mathbf{E}}^{t,\text{text}}] \in \mathbb{R}^{B \times d}. \quad (3.22)$$

The labels are diagonal: query i is paired with candidate i . Thus, each image query must rank its corresponding projected image above all other projected images and all text chunks in the batch, while each text query must rank its corresponding text chunk above all other chunks and all projected images. This directly matches the joint retrieval setting, where both modalities compete in a single ranked list.

The total batch size B grows progressively from 32 to 256 over the first 15 epochs. Smaller batches early in training reduce the number of in-batch negatives, making the ranking task easier when the bridge is not yet well calibrated for the joint setting. As training progresses and alignment improves, the larger candidate pool increases the difficulty of the contrastive objective and sharpens the final ranking.

3.6.5.2 Mixed Candidate Ranking Loss

Stage 4 computes separate raw score blocks for image and text candidates. For a query matrix $\mathbf{Q}$, the uncalibrated image scores are:

$$\mathbf{S}_{\text{img}}^{\text{raw}} = \mathbf{Q} (\mathbf{Z}^{\text{img}})^\top, \quad (3.23)$$

while the text scores are:

$$\mathbf{S}_{\text{text}} = \mathbf{Q} (\tilde{\mathbf{E}}^{t,\text{text}})^\top. \quad (3.24)$$

To calibrate image scores, Stage 4 learns a positive scale $\alpha = \exp(a)$ and an additive bias β . These are applied only to the image score block:

$$\mathbf{S}_{\text{img}} = \alpha \mathbf{S}_{\text{img}}^{\text{raw}} + \beta. \quad (3.25)$$

The final mixed score matrix is then

$$\mathbf{S} = \frac{[\mathbf{S}_{\text{img}}, \mathbf{S}_{\text{text}}]}{\tau}, \quad (3.26)$$

where τ is a fixed ranking temperature used to sharpen the in-batch cross-entropy objective. In contrast to the learned image calibration parameters α and β , τ is not used as a post-hoc score calibration parameter at inference time.

The ranking loss is a cross-entropy loss over the full mixed candidate list:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{B} \sum_{i=1}^B \log \frac{\exp(S_{ii})}{\sum_{j=1}^B \exp(S_{ij})}. \quad (3.27)$$

This is listwise in the practical sense that each query is normalized over the complete mixed image–text candidate list, rather than being evaluated against each candidate independently.

3.6.5.3 Auxiliary Losses

Let $\mathbf{S}_{\text{img,img}}$ denote the calibrated score block restricted to image queries (rows) and projected image candidates (columns). An additional image-transpose loss is applied to this block to encourage symmetry between query-to-image and image-to-query directions:

$$\mathcal{L}_{\text{imgT}} = \text{CE}\left(\frac{\mathbf{S}_{\text{img,img}}^\top}{\tau}, \mathbf{y}_{\text{img}}\right), \quad (3.28)$$

where $\mathbf{y}_{\text{img}}$ is the vector of diagonal labels for image queries.

To reduce the risk that calibrated image scores suppress text retrieval, a text-displacement penalty is added. Let $\mathbf{S}^{\text{txt} \rightarrow \text{img}}$ and $\mathbf{S}^{\text{txt} \rightarrow \text{text}}$ denote the submatrices of $\mathbf{S}$ corresponding to text queries against projected images and text chunks, respectively, and let B_t be the number of text queries in the batch. The displacement loss is

$$\mathcal{L}_{\text{disp}} = \frac{1}{B_t} \sum_{i=1}^{B_t} \max\left(0, \max_j S_{ij}^{\text{txt} \rightarrow \text{img}} - S_{ii}^{\text{txt} \rightarrow \text{text}}\right). \quad (3.29)$$

This term penalizes cases where an image candidate scores above the correct text chunk for a text query.

A collapse regulariser is also used to discourage the bridge from mapping many images to overly similar vectors:

$$\mathcal{L}_{\text{collapse}} = \frac{1}{B_i(B_i - 1)} \sum_{i \neq j} \left(g_\phi(\mathbf{e}_i^{s,\text{img}})^\top g_\phi(\mathbf{e}_j^{s,\text{img}})\right)^2, \quad (3.30)$$

where B_i is the number of image queries in the batch.

Finally, small L_2 penalties are applied to the calibration parameters to prevent unbounded score scaling:

$$\mathcal{L}_{\text{cal}} = \lambda_\alpha a^2 + \lambda_\beta \beta^2. \quad (3.31)$$

The full Stage 4 objective is:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \eta \mathcal{L}_{\text{imgT}} + \lambda_{\text{disp}} \mathcal{L}_{\text{disp}} + \gamma \mathcal{L}_{\text{collapse}} + \mathcal{L}_{\text{cal}}, \quad (3.32)$$

where η , γ , and λ_{disp} are scaling factors to balance the losses.

3.6.5.4 Score Calibration

The learned scale and bias are saved after Stage 4 and applied at inference time only to image scores:

$$s_{\text{img}}(q, x) = \alpha \tilde{\mathbf{q}}^\top g_\phi(\mathbf{e}_x^{s,\text{img}}) + \beta. \quad (3.33)$$

Text scores are left unchanged:

$$s_{\text{text}}(q, c) = \tilde{\mathbf{q}}^\top \tilde{\mathbf{e}}_c^{t,\text{text}}. \quad (3.34)$$

The final joint ranking is produced by sorting the union of calibrated image scores and unmodified text scores. This makes the calibration explicit: the bridge changes the geometry of image vectors, while the learned scale and bias control how strongly projected images compete with text chunks in the final ranked list.

3.7 Experimental Design

This section describes the evaluation conditions, evaluation protocol, and ablation studies used to assess the bridge. All experiments are conducted on the held-out evaluation split produced by the filtering procedure in Section 3.3.4 and reported in Section 4.1.1. Evaluation is performed against the full corpus without filtering or subsampling.

3.7.1 Evaluation Conditions

Five retrieval conditions are evaluated in total. Three are the baselines described in Section 3.4. The remaining two introduce the bridge: a bridge image-only condition, in which an E5 query vector is ranked against bridge-projected image vectors only, providing a direct comparison with the SigLIP image-only baseline; and a bridge joint condition, in which an E5 query vector is ranked against both bridge-projected image vectors and E5 text chunk vectors simultaneously, with image scores calibrated by the learned scale and bias parameters from Stage 4.

3.7.2 Evaluation Protocol

All retrieval conditions are evaluated over the same full corpus. Image retrieval queries are issued against all 427,843 image vectors, and joint retrieval conditions additionally include all 901,571 text chunk vectors in the same ranked list. No filtering or subsampling is applied at evaluation time. This ensures that differences in Hits@ K between conditions reflect differences in retrieval quality rather than differences in corpus coverage or pool composition.

For evaluation, candidate ranking is computed by exhaustive dot-product scoring over vectors loaded from the Qdrant collections. Qdrant is used as the persistent vector store, but the reported metrics are not computed using Qdrant’s approximate HNSW search. This choice isolates the quality of the learned representations and score calibration from the recall–latency tradeoff introduced by approximate nearest-neighbour search. It also makes the comparison between embedding conditions deterministic: each query is scored against the same complete candidate pool. In a deployed system, the same indexed vectors would normally be retrieved through Qdrant’s HNSW index for efficiency.

Retrieval performance is measured using Hits@ K as defined in Section 2.6.1. The evaluation script computes Hits@ K for several diagnostic cutoffs, but the main results report four representative thresholds: top-1, top-0.01% ($K = 43$), top-0.1% ($K = 428$), and top-10% ($K = 42784$). The same absolute K values are used for both image-only and joint retrieval. These values are chosen as percentages of

the 427,843-image corpus so that image retrieval performance remains comparable across conditions.

Results are disaggregated along two axes. First, by query type: `pure_image` queries test retrieval from visual content alone, while `needs_context` queries test retrieval where the question premise is grounded in document context. Second, by image category: results are reported separately for maps, photographs, and charts and diagrams, reflecting the different visual characteristics of each category and their likely different difficulty for cross-modal alignment.

3.7.3 Ablation Study

The ablation study in this thesis varies the number of BridgeBlocks $n \in \{1, 2, 4, 6\}$ to assess the sensitivity of retrieval performance to bridge capacity. All other hyperparameters are held constant and the full pipeline is run for each configuration. This establishes whether the default four-block architecture is necessary or whether a smaller bridge achieves comparable performance.

3.8 Compute

All bridge training was performed on a single NVIDIA T4 GPU (16 GB VRAM) provided by AI Sweden. The full MAPS pipeline (Stages 0–4) completes in approximately 3.5–4 hours on this hardware.

The VQA dataset construction pipeline ran for approximately 75 hours in total. The gatekeeper and judge stages used Qwen2.5-VL-7B-Instruct loaded in 4-bit quantization on the same T4. The generator stage used Mistral Large 3, accessed through AI Sweden’s production inference infrastructure [44]. This infrastructure motivated the choice of Mistral over external commercial APIs.

4

Results

This chapter presents the retrieval performance of the evaluated conditions. The results are organized around four questions: whether the dataset construction pipeline produces a trustworthy evaluation set; whether the bridge resolves cross-modal score incompatibility and improves image retrievability; where in the corpus those improvements are concentrated; and which stages and capacity choices drive the observed gains.

All evaluation results in this chapter are reported as percentages. Unless otherwise stated, image retrieval results are reported at four representative cutoffs: Hits@1, Hits@0.01%, Hits@0.1%, and Hits@10%. Hits@1 is the fraction of evaluation queries for which the gold image is the top 1 retrieved result, and more generally Hits@ $N\%$ is the fraction of queries for which the gold image appears within the top $N\%$ of the ranked candidate list.

4.1 Dataset Construction Results

Before reporting retrieval performance, it is necessary to establish that the evaluation data are sufficiently clean to support meaningful conclusions. This section reports the output of the dataset construction pipeline in terms of record counts, category distribution, and estimated routing noise.

4.1.1 Dataset Counts

To establish how many examples survive each stage of the dataset construction pipeline, Table 4.1 summarises the number of records retained at each stage, reported separately for image-based VQA triplets and text QA pairs. In the text QA pipeline, records that did not pass the judge filter were discarded during generation, meaning that only gold-filtered text QA pairs were retained for later splitting and evaluation.

The protected evaluation splits are held out before bridge training and are never used for optimization, model selection, or calibration. All text QA records that failed any gold-filtering criterion, a judge score below 3, hallucination markers in the judge reasoning, a non-empty error field, or a missing required field (question, answer, judge reasoning, or chunk text), were discarded during generation and not tracked, so no text QA records were removed at the gold-filtering stage.

The distribution of these gold VQA records across the three visual categories and two query types is reported in Table 4.2 and examined in Section 4.1.2, once routing noise has been established.

Table 4.1: Dataset counts after generation, judging, filtering, and splitting. The generated count is not reported for Text QA because non-gold records were discarded and not tracked during generation.

Stage	VQA triplets	Text QA pairs
Generated records	34,492	–
Gold records (score ≥ 3)	22,509	3,341
Protected evaluation split	1,398	500
Training split	21,111	2,841

Table 4.2: Distribution of gold VQA records across the three visual categories and two query types.

Category	pure_image	needs_context	Total
maps	2,223	5,279	7,502
photos	4,802	2,704	7,506
charts_diagrams	5,820	1,681	7,501
Total	12,845	9,664	22,509

4.1.2 Image Routing Noise

To assess how reliably images were routed into the three visual categories, manual inspection of 50 randomly sampled images per category was carried out. Table 4.3 reports the estimated routing noise.

Table 4.3: Estimated image routing noise by category. Noise rate is the fraction of sampled images judged to belong to a different category. 95% Wilson score confidence intervals computed from $n = 50$ samples per category; the Wilson interval is used because the normal-approximation interval is degenerate at the zero misroutes observed for **maps**.

Category	Sampled	Misrouted	Noise rate	95% CI
maps	50	0	0%	0–7%
charts_diagrams	50	28	56%	42–69%
photos	50	32	64%	50–76%

The **maps** category shows zero misroutes in the sample. The **charts_diagrams** and **photos** categories show noise rates of 56% and 64% respectively. Manual inspection revealed that misrouted images in both categories consisted predominantly of map images. No images from the excluded categories—logos, stamps, or text scans—were found in the sample.

The high noise rates in **charts_diagrams** and **photos** mean that these two category labels are not reliable as semantic categories in the evaluation data. Category-level retrieval results for these two categories should therefore be interpreted with caution, and the **maps** category result is the most interpretable of the three.

With routing noise established, the gold VQA training counts in Table 4.2 can be interpreted. The three categories are balanced in total size, each contributing approximately 7,500 training records. Within categories, the query type distribution differs substantially. `maps` records are predominantly `needs_context` (70%), reflecting that municipal maps typically require surrounding document text to be interpreted correctly. By contrast, `charts_diagrams` records are predominantly `pure_image` (78%), reflecting that charts and diagrams more often carry self-contained visual information.

4.2 Retrieval Results

This section reports the main retrieval results across the image-only, joint image-text, and text retrieval settings. The results evaluate whether the learned bridge makes images retrievable through text queries, whether it enables images and text to be ranked together, and whether adding images affects the existing E5 text retrieval pipeline.

4.2.1 Baseline Score Incompatibility

Before evaluating retrieval performance under the bridge, it is worth establishing whether unaligned embedding spaces can be combined at all. Table 4.4 reports the score distributions produced by the two baseline joint conditions, which merge independently trained embedding spaces without alignment.

Table 4.4: Text and image score distributions for the baseline joint conditions. Score means are computed over all queries and candidates in the evaluation pool.

Condition	Text mean	Image mean
Naive joint (E5 text + SigLIP images)	0.7810	0.0462
SigLIP multimodal	0.5605	0.0462

In the naive joint condition, E5 text scores have a mean of 0.7810 while SigLIP image scores have a mean of only 0.0462. Because all image scores fall far below all text scores, every position in the joint ranked list is occupied by a text candidate, and no image appears in any top- K result at any reported cutoff. The SigLIP multimodal condition exhibits the same failure: although SigLIP is a jointly trained vision-language model, its text scores (mean 0.5605) still dominate its image scores completely. The score incompatibility problem is not resolved by using a purpose-built vision-language encoder when its text and image encoders produce scores on different scales. The consequence of this incompatibility is reflected directly in Table 4.6: both baseline joint conditions achieve 0.00% at every reported cutoff, while the bridge joint condition achieves 25.39% Hits@0.01% by mapping image scores into the same space as text scores.

4.2.2 Image-Only Retrieval

This subsection asks whether the bridge improves image retrievability when only images are ranked. Table 4.5 reports image retrieval performance for the SigLIP image-only baseline and the bridge image-only condition, both ranked over all images in the dataset. The bridge image-only condition uses an E5 query encoder, while SigLIP image-only uses a SigLIP text query encoder.

Table 4.5: Image-only retrieval results. Both conditions rank over all images in the dataset. The gray row is the bridge condition. Bold denotes the best value per column.

Condition	Hits@1	Hits@0.01%	Hits@0.1%	Hits@10%
SigLIP image-only	1.65	6.80	13.23	57.37
Bridge image-only	2.93	26.90	48.64	93.63

The bridge substantially improves image retrievability across all cutoffs. Hits@0.01% increases from 6.80% to 26.90%, and Hits@0.1% increases from 13.23% to 48.64%. At the broadest cutoff, Hits@10% rises from 57.37% to 93.63%, indicating that the bridge places the large majority of relevant images within a practically inspectable candidate set.

4.2.3 Joint Image–Text Retrieval

The image-only gains of Section 4.2.2 are only useful if the bridge also enables images to be ranked together with text. Table 4.6 reports retrieval performance in the joint setting, where image candidates compete against all text chunks in the dataset in a single ranked list.

Table 4.6: Joint image retrieval results. All conditions rank over all images and text chunks in the dataset simultaneously. The gray row indicates the bridge condition where images and text are scored in a compatible space. Bold denotes the best value per column.

Condition	Hits@1	Hits@0.01%	Hits@0.1%	Hits@10%
Naive joint (E5 text + SigLIP images)	0.00	0.00	0.00	0.00
SigLIP multimodal	0.00	0.00	0.00	0.00
Bridge joint	2.58	25.39	42.78	59.37

The two baseline joint conditions fail completely, as explained by the score distributions in Section 4.2.1. The bridge joint condition retains most of the image-only gains from Table 4.5. Hits@0.01% decreases only from 26.90% to 25.39%, and Hits@0.1% from 48.64% to 42.78%, despite ranking against all text chunks in the dataset. Hits@10% shows a larger drop from 93.63% to 59.37%, reflecting increased competition from the text corpus at broad cutoffs.

4.2.4 Text Retrieval and Displacement

Adding bridge-projected images to the text index could displace relevant text chunks from the top of the ranking, undermining the existing E5 retrieval pipeline. The bridge outputs L2-normalized vectors that live in the centered E5 space established in MAPS’ Stage 0 (Section 3.6.1), and at query time the E5 query is centered before scoring; the joint index is therefore evaluated entirely in this centered space. The appropriate reference is thus the centered E5 text-only baseline rather than the raw E5 pipeline. Comparing against the uncentered baseline would conflate two distinct effects: the shift caused by mean-centering the text space, which is a prerequisite of the bridge setup, and the displacement caused by inserting image vectors into the index. Measuring deltas from the centered baseline isolates the latter. Table 4.7 quantifies this displacement on the held-out 500-record text QA evaluation set, expressed as deltas from the centered E5 text-only baseline.

Table 4.7: Text retrieval on the held-out 500-record text QA evaluation set, shown as deltas (Δ) from the centered E5 text-only baseline. Δ = condition – centered baseline: negative values indicate a decrease relative to the centered baseline (displacement), positive values an improvement. The text-joint condition additionally includes bridge-projected image vectors. The gray row indicates the bridge performance.

Condition	Hits@1	Hits@0.01%	Hits@0.1%	Hits@10%
E5 text-only (centered, baseline)	36.63	76.07	87.92	98.78
Δ E5 + bridge joint	−12.42	−4.37	−0.47	−0.08

Adding bridge-projected images to the centered text index causes a large drop at Hits@1 (−12.42) and a moderate drop at Hits@0.01% (−4.37), but the effect shrinks rapidly at broader cutoffs, to −0.47 at Hits@0.1% and −0.08 at Hits@10%. This indicates that displacement is concentrated near the very top of the ranking, while broad-cutoff text retrieval is essentially preserved.

4.3 Category and Query Type Analysis

This section analyzes whether retrieval performance differs across visual categories and query types. Because the automatic category labels contain substantial routing noise, the category-level results should be interpreted as approximate rather than as clear estimates of performance for each visual class. The query type analysis is therefore used as a complementary view of where the bridge improves retrieval.

4.3.1 Category-Level Performance

To localise where the bridge’s gains come from, image-only retrieval is broken down across three subsets: `maps` only, `maps` combined with `charts_diagrams`, and `maps` combined with `photos`. Table 4.8 compares the SigLIP image-only baseline (S)

with the bridge image-only condition (B), with bridge columns shaded. Because the routing-noise analysis in Table 4.3 showed that many `charts_diagrams` and `photos` records were map-like, these subsets should be interpreted as progressively noisier map-related groups rather than clean visual categories.

Table 4.8: Category-level image retrieval results for the SigLIP image-only baseline (S) and bridge image-only condition (B). Bridge columns are shaded. Bold denotes the best value per column.

Category	Hits@1		Hits@0.01%		Hits@10%	
	S	B	S	B	S	B
<code>maps + charts_diagrams</code>	4.09	5.17	16.38	44.18	72.63	94.83
<code>maps + photos</code>	0.81	2.24	2.85	17.89	58.33	93.09
<code>maps</code>	0.00	1.36	1.13	18.78	40.27	92.99

The bridge improves retrieval across all three subsets. For the `maps + charts_diagrams` subset, SigLIP achieves the strongest result at Hits@0.01% (16.38%), and the bridge more than doubles this to 44.18%. For the `maps` and `maps + photos` subsets, the SigLIP baseline is near zero at strict cutoffs; the bridge raises both to around 18% Hits@0.01% and above 90% Hits@10%.

The `maps` result is the most interpretable given the zero routing noise established in Section 4.1.2. SigLIP Hits@0.01% for `maps` is 1.13%, rising to 18.78% for the bridge. At the broadest cutoff, the bridge raises `maps` Hits@10% from 40.27% to 92.99%.

4.3.2 Query Type Analysis

Whether the bridge’s gains stem from queries that can be answered from the image alone, or from queries that require surrounding document context, can be examined by splitting evaluation queries by their query type. Table 4.9 reports image retrieval broken down by query type for the SigLIP baseline and bridge image-only condition. Bridge columns are shaded.

Table 4.9: Image retrieval by query type for the SigLIP image-only baseline (S) and bridge image-only condition (B). Bridge columns are shaded. Bold denotes the best value per column.

Query type	Hits@0.01%		Hits@0.1%	
	S	B	S	B
<code>pure_image</code>	11.11	29.43	20.63	47.25
<code>needs_context</code>	0.69	23.32	2.76	50.60

SigLIP performs substantially better on `pure_image` queries than on `needs_context` queries. The bridge narrows this gap: Hits@0.01% rises from 0.69% to 23.32% for `needs_context` queries, and from 11.11% to 29.43% for `pure_image` queries. At

Hits@0.1%, the `needs_context` bridge result (50.60%) slightly exceeds the `pure_image` result (47.25%).

4.4 Ablation Study

This section examines the effect of bridge capacity on retrieval performance. The ablation compares bridge variants with different numbers of BridgeBlocks, measuring both image retrieval gains and the extent to which bridge-projected images displace text candidates in the joint index.

4.4.1 Bridge Capacity

To examine how much representational capacity the bridge needs, image-only retrieval is measured as a function of the number of BridgeBlocks. Table 4.10 reports the results after the full MAPS pipeline including Stage 4 calibration.

Table 4.10: Bridge capacity ablation in the image-only setting after Stage 4 calibration. Parameter counts exclude the frozen SigLIP and E5 encoders. The gray row is the final bridge used in all other experiments. Bold denotes the best value per column.

Configuration	Parameters	Hits@1	Hits@0.01%	Hits@0.1%
1 BridgeBlock	5.4M	2.22	21.82	42.06
2 BridgeBlocks	9.6M	2.15	21.17	41.42
4 BridgeBlocks	18.0M	2.93	26.90	48.64
6 BridgeBlocks	26.4M	3.08	23.68	44.92

The 4-block configuration achieves the best performance at Hits@0.01% and Hits@0.1%. The 6-block model achieves slightly higher Hits@1 but lower medium-cutoff performance. The 1- and 2-block models perform similarly to each other and below the 4-block model at all reported cutoffs.

4.4.2 Joint Retrieval and Text Displacement

Capacity choices that look strong in isolation may behave differently in the joint index, where bridge variants must compete with text. Table 4.11 reports image retrieval in the joint setting for each bridge capacity, with images and text chunks competing in the same ranked list.

The 4-block bridge achieves the strongest joint retrieval at Hits@0.01%, Hits@0.1%, and Hits@10%. The 6-block model achieves slightly higher Hits@1 in the joint setting but lower performance at broader cutoffs.

To quantify how strongly each bridge capacity displaces text candidates, Table 4.12 reports text retrieval as deltas from the centered text-only baseline.

All bridge variants cause some decrease in strict top-rank text retrieval relative to the centered text-only baseline. At broader cutoffs, text retrieval is largely preserved

Table 4.11: Bridge capacity ablation under joint retrieval after Stage 4 calibration. Images compete directly with all text chunks in the dataset. The gray row is the final bridge used in this thesis. Bold denotes the best value per column.

Configuration	Hits@1	Hits@0.01%	Hits@0.1%	Hits@10%
1 BridgeBlock	2.07	19.10	33.76	50.36
2 BridgeBlocks	1.93	19.17	33.83	46.99
4 BridgeBlocks	2.58	25.39	42.78	59.37
6 BridgeBlocks	2.79	21.53	37.34	51.93

Table 4.12: Text retrieval displacement under joint retrieval for each bridge capacity, shown as deltas (Δ) from the centered E5 text-only baseline. $\Delta = \text{condition} - \text{baseline}$: negative values indicate displacement of text by bridge-projected images. The dark gray row is the reference baseline (raw values), and the light gray row is the final 4-block bridge used in this thesis.

Condition	Hits@1	Hits@0.01%	Hits@0.1%	Hits@10%
E5 text-only (centered, baseline)	36.63	76.07	87.92	98.78
Δ 1 BridgeBlock	-8.56	-1.12	-0.05	0.00
Δ 2 BridgeBlocks	-7.16	-0.91	0.00	0.00
Δ 4 BridgeBlocks	-12.42	-4.37	-0.47	-0.08
Δ 6 BridgeBlocks	-4.63	-0.49	-0.04	0.00

across all configurations. The 4-block bridge shows more displacement at Hits@1 than the smaller variants, reflecting that its stronger image calibration places more image candidates above the gold text chunk at the very top of the ranking. At Hits@0.1% and beyond, all variants preserve text retrieval almost identically.

Taken together, these results point to a trade-off rather than a clean capacity trend: the 4-block bridge achieves the strongest joint image retrieval (Table ??) but displaces the most text at the very top of the ranking, while displacement at broader cutoffs is small and does not increase monotonically with capacity.

5

Conclusion

5.1 Summary of Findings

This thesis investigated whether embedding-space alignment can extend an existing text-based RAG retrieval pipeline to visual document elements without re-embedding the text corpus. The results confirm that it can, as a proof of concept. The learned bridge substantially improves image retrievability and makes joint image–text ranking possible, while also revealing a remaining trade-off between visual retrieval and strict top-rank text retrieval.

The first research question asked:

Can Vec2Vec mappings achieve retrieval-compatible alignment between independently trained text and image embedding spaces?

The results show that direct score merging is not viable: both the naive E5+SigLIP joint condition and the SigLIP multimodal condition retrieved no gold images at any reported cutoff, indicating that score incompatibility is a central obstacle rather than a minor calibration issue. Using a purpose-built vision–language model does not automatically resolve this problem. After alignment, however, the bridge reached 25.39% Hits@0.01% in the joint setting and 26.90% in the image-only setting, with Hits@10% reaching 93.63% in the image-only condition. The learned mapping therefore achieves retrieval-compatible alignment in the E5 space: it makes visual evidence substantially more retrievable under the same text-oriented scoring function.

The second research question asked:

Which approach - Vec2Vec alignment or unified multimodal embeddings, in this thesis: SigLIP - demonstrates better retrieval effectiveness on mixed-media document collections?

In this experimental setting, the learned bridge was the stronger approach for retrieving visual evidence in the existing E5-based retrieval pipeline. The unified SigLIP multimodal baseline failed to surface images in joint retrieval, while the bridge reached 25.39% image Hits@0.01% when images and text were ranked together. For this corpus and retrieval setup, post-hoc alignment into the established text space was therefore more effective than directly merging independently scored multimodal embeddings.

The category and query-type analyses clarify where this advantage arises, but they should be interpreted in light of the routing noise described in Section 4.1.2.

The strongest apparent gains occur for `maps` and `needs_context` queries, which are precisely the cases where visual content is difficult to retrieve using direct SigLIP similarity alone. `maps` improved from 1.13% to 18.78% Hits@0.01%, and `needs_context` queries improved from 0.69% to 23.32%. This suggests that the bridge is particularly useful for visual evidence whose meaning depends on the surrounding document context. However, because the automatic category labels for `photos` and `charts_diagrams` carry substantial noise and many images in these groups are in fact maps, the exact magnitudes of the category-level gains should be viewed as indicative rather than definitive. The improvement is also not cost-free: strict text Hits@1 decreased from 36.63% (centered E5 baseline) to 24.21% in the joint setting, although broader text retrieval at Hits@0.1% was largely preserved, changing only from 87.92% to 87.45%.

5.2 Contributions

The thesis offers three principal contributions.

First, it introduces MAPS, a staged training pipeline that combines target-space shaping, text-to-text pretraining, paired image–query anchoring, corpus extension with pseudo-labels, and unified multimodal ranking with learned score calibration. A key practical property of MAPS is that the text encoder and existing text index remain fixed throughout; visual support is added by learning a retrieval-stage mapping rather than by replacing the embedding infrastructure.

Second, it constructs a supervised evaluation resource for visual document retrieval over Swedish municipal PDFs. The automated VQA pipeline produced 22,509 gold VQA triplets and 3,341 text QA pairs, with protected evaluation splits held out from all training and calibration. This made it possible to evaluate retrieval directly against image and text evidence in a domain where no benchmark was previously available.

Third, the thesis provides an empirical comparison between post-hoc alignment into an existing E5 retrieval space and SigLIP-based multimodal retrieval baselines on a large real-world mixed-media corpus, offering evidence-based guidance on when vector-space alignment is a viable strategy for extending text retrieval pipelines to multimodal documents.

5.3 Practical Implications

The approach studied here is practically attractive when an organisation already maintains a large text-based vector index. Replacing the text encoder with a fully multimodal model would require re-embedding and re-indexing the entire corpus, an operation that is expensive and operationally disruptive for large archives such as the corpus considered in the thesis, consisting of 235,000 municipal PDF documents. In contrast, the bridge keeps the existing E5 text space fixed and adds visual retrieval at the retrieval stage only. If new documents are added, their text chunks can continue to be indexed with E5 and their images projected through the existing bridge. The

bridge would need to be retrained only if the target text embedding model itself changes.

Because the target space is multilingual-e5-large, the approach is not restricted to Swedish. The same pipeline could be applied to other non-English municipal document corpora where a text index already exists, without modifications to the bridge architecture or training procedure.

The VQA generation pipeline represents a one-time investment that produces a reusable evaluation resource. The approximately 75-hour generation cost yields a gold evaluation dataset that can be used to assess future retrieval improvements without repeating the full generation process. This is relevant for organisations that need domain-specific evaluation data but lack annotated multimodal retrieval benchmarks.

The findings are consistent with prior work on embedding-space translation, which shows that retrieval-useful structure can survive a learned mapping between independently trained embedding spaces [18], [19]. The multimodal setting is harder than the text-to-text case studied in Vec2Vec because images and passages are not two encodings of the same object; they provide partially overlapping evidence about the same underlying content. The complete failure of the SigLIP multimodal baseline supports the theoretical observation that score compatibility is a learned geometric property rather than a consequence of joint training alone [35]. These results contrast with full visual-document retrieval approaches such as ColPali-style page-level multimodal indexing [14], which may offer stronger visual retrieval but require a different indexing strategy and cannot easily coexist with an existing text-based pipeline.

5.4 Limitations

Several limitations constrain the conclusions that can be drawn from this work.

All experiments were conducted on a single domain: Swedish municipal PDF documents. The MAPS pipeline and the VQA evaluation dataset were both constructed from this corpus. It is therefore unclear how well the bridge generalises to other document types, visual styles, or languages. The improvement observed for **maps** in particular may reflect properties of municipal cartographic content that do not transfer to other visual domains such as scientific figures, medical imaging, or product catalogues.

The VQA dataset was generated and filtered automatically using an LLM-as-judge pipeline. Despite gold filtering and manual noise estimation, some proportion of evaluation pairs will contain errors. The routing-noise analysis further showed that the **charts_diagrams** and **photos** category labels carry noise rates of 56% and 64% respectively, with misrouted images consisting predominantly of maps. Category-level conclusions for these two categories should therefore be treated as indicative rather than definitive.

The corpus extension in Stage 3 relies on pseudo-queries derived from surrounding document text. The confidence gate used to filter low-quality pseudo-labels is heuristic and based on cosine similarity to known VQA queries. The semantic quality of retained pseudo-labels is not independently verified, and noisy pseudo-labels may

have introduced misalignment for images whose document context is uninformative or misleading.

All training was conducted on a single NVIDIA T4 GPU with 16 GB of VRAM, limiting feasible batch sizes and bridge capacity. More fundamentally, the MAPS pipeline depends on a domain-specific VQA dataset to anchor the bridge in Stage 2. Constructing this dataset required a purpose-built generation pipeline running for approximately 75 hours, producing 22,509 gold pairs after filtering. This annotation cost is a practical barrier to applying the approach in new domains: a new corpus requires a new VQA dataset before any bridge training can begin. The pipeline is therefore not zero-shot transferable, and its scalability is constrained by the cost of supervised pair construction.

The text displacement at Hits@1 is a genuine limitation of the current calibration. Strict top-rank text retrieval decreases from 36.63% (centered E5 baseline) to 24.21% in the joint setting, a drop of over 12 percentage points, which may be unacceptable in applications where the first retrieved result must be a text chunk. The Stage 4 text-displacement penalty reduces but does not eliminate this effect.

It is worth noting that the Stage 4 calibration is optimised for ranking rather than distribution alignment. The learned scale and bias are fitted to make relevant image scores competitive with text scores in the ranked list, not to match the marginal score distributions of the two modalities. As a consequence, the calibrated image score distribution does not necessarily resemble the text score distribution in shape or spread; it is only constrained to place gold images above the text score threshold at which they need to rank. The bridge may therefore succeed at retrieval while producing image scores whose absolute values look incomparable to text scores in isolation.

Finally, the evaluation measures retrieval quality only. It does not measure whether a downstream language model produces better final answers when given the retrieved images alongside text evidence. Retrieval improvements at the corpus scale evaluated here do not automatically translate to downstream answer quality, and an end-to-end RAG evaluation may show a different picture.

5.5 Future Work

Several directions follow directly from the limitations above.

Testing the bridge on document collections from other domains and languages would establish whether the learned alignment is specific to municipal Swedish documents or whether it generalises more broadly. Negative transfer, where bridge training on one domain degrades performance on another, is a concrete risk that has not been assessed.

The current Stage 3 pseudo-label construction is heuristic and dependent on the quality of surrounding document text. Replacing it with a stronger multimodal model to generate image descriptions, or using human-verified image-text pairs for a subset of the corpus, could improve the quality of supervision for unannotated images and reduce alignment noise.

The bridge improves candidate retrievability more strongly than strict top-rank accuracy. A lightweight reranker applied to the top- K bridge results could improve

Hits@1 without changing the bridge itself. Connecting retrieval to end-to-end RAG evaluation—for example, by measuring whether improved visual retrieval leads to better generated answers, or whether retrieved images are judged useful by human evaluators—would give a more complete picture of practical value.

Extending the VQA generation pipeline to cover a larger and more representative fraction of corpus images would increase in-batch negative diversity and likely improve both alignment quality and calibration. Applying Stage 4 score calibration to the SigLIP multimodal condition and comparing it to the bridge under matched conditions would clarify whether the bridge advantage comes from the learned mapping or from calibration alone. Comparisons against modern unified multimodal embedding models such as Gemini Embedding 2 [34] that require full corpus re-indexing would also quantify the cost–benefit trade-off between the bridge approach and a complete infrastructure replacement.

Extending MAPS toward an unsupervised regime—where the bridge is trained using only unpaired corpus images and text chunks, without any annotated image–query pairs—would remove the VQA construction bottleneck entirely and make the approach applicable to any corpus with an existing text index. The universal geometry hypothesis [19] suggests that independently trained embedding models of sufficient capacity tend to converge on shared latent structure, and Dar [20] has shown that this structure is strong enough to support alignment with linear transformations alone. Whether the cross-modal case studied here, where images and text encode partially overlapping rather than identical information, retains enough shared geometric structure for unsupervised alignment to succeed is an open and practically important question.

Finally, continual learning—updating the bridge incrementally as new documents are added without full retraining—is a further direction aligned with the operational deployment context of large municipal document archives.

5.6 Final Remarks

The bridge substantially improves the probability that relevant visual evidence appears within a candidate set, particularly at medium and broad cutoffs. The improvement is less decisive at Hits@1, which suggests that the bridge is better suited as a candidate-generation mechanism than as a final ranking model, and that a reranking step would likely improve strict top-rank performance. The results should therefore be interpreted as a successful proof of concept rather than as a production-ready retrieval system.

This thesis demonstrates that embedding-space alignment can be a viable way to extend text-based RAG retrieval with visual document evidence. The learned bridge does not remove all ranking and calibration challenges, but it changes the problem from one where images are effectively invisible in joint retrieval to one where visual evidence can compete in the same ranked list as text. For the Swedish municipal document corpus studied here, this means that visual content, particularly maps and other context-dependent document images, can be surfaced by natural language queries alongside relevant text passages. The approach therefore offers a practical path for organisations that need to extend existing retrieval infrastructure to multi-

modal content without the cost and disruption of full corpus re-indexing. Whether the supervised anchor signal that currently drives this alignment can eventually be replaced by latent geometric structure shared between independently trained encoders remains an open question, and one whose answer will determine how broadly the approach can scale.

Code Availability

The code for this thesis is available on request from the author.

Bibliography

- [1] OpenAI. “Introducing ChatGPT search.” Accessed: 2026-03-18. [Online]. Available: <https://openai.com/index/introducing-chatgpt-search/>.
- [2] Google. “AI Overviews in search.” Accessed: 2026-03-18. [Online]. Available: <https://blog.google/products/search>.
- [3] Z. Ji et al., “Survey of hallucination in natural language generation,” *ACM Computing Surveys*, vol. 55, no. 12, 2022. DOI: 10.1145/3571730.
- [4] L. Huang et al., “A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions,” *ACM Transactions on Information Systems*, 2025. DOI: 10.1145/3703155. [Online]. Available: <https://arxiv.org/abs/2311.05232>.
- [5] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474. arXiv: 2005.11401 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2005.11401>.
- [6] Y. Gao et al., “Retrieval-augmented generation for large language models: A survey,” *arXiv preprint arXiv:2312.10997*, Mar. 2024. arXiv: 2312.10997 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2312.10997>.
- [7] Y. Huang and J. X. Huang, “A survey on retrieval-augmented text generation for large language models,” *arXiv preprint arXiv:2404.10981*, 2024. arXiv: 2404.10981 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2404.10981>.
- [8] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, “Self-RAG: Learning to retrieve, generate, and critique through self-reflection,” *arXiv preprint arXiv:2310.11511*, 2023. arXiv: 2310.11511 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2310.11511>.
- [9] M. M. Abootorabi et al., “Ask in any modality: A comprehensive survey on multimodal retrieval-augmented generation,” *arXiv preprint arXiv:2502.08826*, 2025. arXiv: 2502.08826 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2502.08826>.
- [10] L. Mei, S. Mo, Z. Yang, and C. Chen, “A survey of multimodal retrieval-augmented generation,” *arXiv preprint arXiv:2504.08748*, 2025. arXiv: 2504.08748 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2504.08748>.

- [11] A. Radford et al., “Learning transferable visual models from natural language supervision,” in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139, 2021, pp. 8748–8763. [Online]. Available: <https://arxiv.org/abs/2103.00020>.
- [12] J.-B. Alayrac et al., “Flamingo: A visual language model for few-shot learning,” *arXiv preprint arXiv:2204.14198*, 2022. arXiv: 2204.14198 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2204.14198>.
- [13] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” *arXiv preprint arXiv:2304.08485*, 2023. arXiv: 2304.08485 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2304.08485>.
- [14] M. Faysse et al., “ColPali: Efficient document retrieval with vision language models,” *arXiv preprint arXiv:2407.01449*, 2024. arXiv: 2407.01449 [cs.IR]. [Online]. Available: <https://arxiv.org/abs/2407.01449>.
- [15] C. Barboule, B. Piwowarski, and Y. Chabot, “Survey on question answering over visually rich documents: Methods, challenges, and trends,” *arXiv preprint arXiv:2501.02235*, 2025. arXiv: 2501.02235 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2501.02235>.
- [16] D. Wang et al., “DocLLM: A layout-aware generative language model for multimodal document understanding,” in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, 2024, pp. 8529–8548. [Online]. Available: <https://aclanthology.org/2024.acl-long.463/>.
- [17] L. Wang et al., “Text embeddings by weakly-supervised contrastive pre-training,” *arXiv preprint arXiv:2212.03533*, 2022. arXiv: 2212.03533 [cs.CL]. [Online]. Available: <https://arxiv.org/abs/2212.03533>.
- [18] A. K. Gao, “Vec2vec: A compact neural network approach for transforming text embeddings with high fidelity,” *arXiv preprint arXiv:2306.12689*, 2023. arXiv: 2306.12689 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2306.12689>.
- [19] R. Jha, C. Zhang, V. Shmatikov, and J. X. Morris, “Harnessing the universal geometry of embeddings,” in *Advances in Neural Information Processing Systems*, 2025. arXiv: 2505.12540 [cs.CL]. [Online]. Available: <https://openreview.net/forum?id=jiCLUPq5xv>.
- [20] G. Dar, “Mini-vec2vec: Scaling universal geometry alignment with linear transformations,” *arXiv preprint arXiv:2510.02348*, 2025. arXiv: 2510.02348 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2510.02348>.
- [21] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 3rd. 2026, Online manuscript, released 2026-01-06; accessed: 2026-03-19. [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/>.

- [22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *Proceedings of the 1st International Conference on Learning Representations, Workshop Track*, 2013. [Online]. Available: <https://arxiv.org/abs/1301.3781>.
- [23] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using siamese BERT-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410.
- [24] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://arxiv.org/abs/2010.11929>.
- [25] C. Jia et al., “Scaling up visual and vision-language representation learning with noisy text supervision,” in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139, 2021, pp. 4904–4916. [Online]. Available: <https://arxiv.org/abs/2102.05918>.
- [26] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, “Sigmoid loss for language image pre-training,” *arXiv preprint arXiv:2303.15343*, 2023. arXiv: 2303.15343 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2303.15343>.
- [27] A. Vaswani et al., “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Online]. Available: <https://arxiv.org/abs/1706.03762>.
- [28] M. Geva, R. Schuster, J. Berant, and O. Levy, “Transformer feed-forward layers are key-value memories,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 5484–5495. DOI: 10.18653/v1/2021.emnlp-main.446.
- [29] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training,” *OpenAI technical report*, 2018. [Online]. Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.
- [30] T. B. Brown et al., “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.
- [31] V. Karpukhin et al., “Dense passage retrieval for open-domain question answering,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online: Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. [Online]. Available: <https://aclanthology.org/2020.emnlp-main.550>.
- [32] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford, “Okapi at TREC-3,” in *Proceedings of the Third Text REtrieval Conference (TREC-3)*, 1994.

- [33] Y. A. Malkov and D. A. Yashunin, “Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 824–836, 2020. DOI: 10.1109/TPAMI.2018.2889473. [Online]. Available: <https://arxiv.org/abs/1603.09320>.
- [34] Google DeepMind and Google Research. “Gemini Embedding 2: Our first natively multimodal embedding model.” Accessed: 2026-03-18. [Online]. Available: <https://blog.google/innovation-and-ai/models-and-research/gemini-embedding-2/>.
- [35] V. W. Liang, Y. Zhang, Y. Kwon, S. Yeung, and J. Y. Zou, “Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 612–17 625, 2022. [Online]. Available: <https://arxiv.org/abs/2203.02053>.
- [36] M. Huh, B. Cheung, T. Wang, and P. Isola, “Position: The Platonic Representation Hypothesis,” in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 235, PMLR, 2024, pp. 20 617–20 642. [Online]. Available: <https://proceedings.mlr.press/v235/huh24a.html>.
- [37] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008. [Online]. Available: <https://nlp.stanford.edu/IR-book/>.
- [38] S. Antol et al., “VQA: Visual question answering,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2425–2433. DOI: 10.1109/ICCV.2015.279.
- [39] K. Marino, M. Rastegari, A. Farhadi, and R. Mottaghi, “OK-VQA: A visual question answering benchmark requiring external knowledge,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3195–3204. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2019/html/Marino_OK-VQA_A_Visual_Question_Answering_Benchmark_Requiring_External_Knowledge_CVPR_2019_paper.html.
- [40] AI Sweden, *En gemensam digital assistent för offentlig sektor*, <https://www.ai.se/sv/projekt/en-gemensam-digital-assistent-offentlig-sektor>, Accessed 21 May 2026, 2024.
- [41] LlamaIndex. “LlamaIndex documentation.” Accessed: 2026-03-18. [Online]. Available: <https://docs.llamaindex.ai/>.
- [42] Qdrant. “Qdrant documentation.” Accessed: 2026-03-18. [Online]. Available: <https://qdrant.tech/documentation/>.
- [43] S. Bai et al., “Qwen2.5-VL technical report,” *arXiv preprint arXiv:2502.13923*, 2025. arXiv: 2502.13923 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2502.13923>.
- [44] Mistral AI. “Mistral Large 3.” Accessed: 2026-03-18. [Online]. Available: <https://docs.mistral.ai/models/model-cards/mistral-large-3-25-12>.

- [45] J. Wei et al., “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.
- [46] A. Machina and R. Mercer, “Anisotropy is not inherent to transformers,” in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Mexico City, Mexico: Association for Computational Linguistics, Jun. 2024, pp. 4892–4907. [Online]. Available: <https://aclanthology.org/2024.naacl-long.274/>.
- [47] J. Kirkpatrick et al., “Overcoming catastrophic forgetting in neural networks,” *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017. DOI: 10 . 1073 / pnas . 1611835114. [Online]. Available: [https : //arxiv.org/abs/1612.00796](https://arxiv.org/abs/1612.00796).

A

Appendix A: Generation Prompts

A.1 VQA Gatekeeper Prompt

The following prompt is sent to Qwen2.5-VL-7B-Instruct together with each image. The system message instructs the model to respond only with valid JSON. The user prompt is in Swedish to match the language of the corpus.

System Message

Du är en AI-assistent. Svara endast med giltig JSON.

User Prompt

Du är en expert på visuell dataanalys. Din uppgift är att utvärdera en bild helt BLIND (utan extern text) och avgöra dess klassificering.

REGLER:

1. "useless": Bilden är en logotyp, ett gränssnittselement (UI), ett fotografi utan sökbar data, rent dekorativ, eller saknar information.
2. "needs_context": Bilden innehåller data (kartor, diagram) men har specifika markeringar (t.ex. en röd linje), okända symboler, eller saknar en teckenförklaring som gör att den kräver ett dokument för att förstås helt.
3. "pure_image": Bilden är ett självförklarande diagram, tabell eller karta som kan förstås till 100% utan text.

Returnera ENDAST en JSON med nyckeln "query_type".

Exempel: {"query_type": "pure_image"}

A.2 VQA Generator Prompts

Two generator prompts are used depending on the query type assigned by the gatekeeper. Both are sent to Mistral Large 3 together with the image. All prompts are in Swedish to match the language of the corpus. The response format is enforced as JSON with temperature 0.0.

Generator Prompt — pure_image

Du är en expert på att skapa träningsdata för Multimodal QA (Question Answering).

Din uppgift är att skriva en fråga och ett svar baserat ENBART på den bifogade bilden. Du har INTE tillgång till någon extern text.

REGLER:

1. Självbärande fråga (Blind användare): Frågan ska låta som en naturlig sökning. Användaren vet INTE att det finns en bild. FÖRBJUDET att använda syftningar som "i bilden", "den röda stapeln", eller "kartan visar".
2. Explicit förankring: Det genererade svaret måste kunna avläsas exakt och otvetydigt från bildens pixlar. Hitta inte på värden.

UPPGIFT:

Returnera din bedömning i JSON-format.

```
{
  "target_insight": "<Kort beskrivning av den specifika
                    datapunkt du fokuserar på>",
  "question": "<Din självbärande sökfråga på SVENSKA>",
  "answer": "<Det exakta svaret avläst från bilden på SVENSKA>",
  "explanation": "<Steg-för-steg bevis på hur bilden visar
                svaret på SVENSKA>"
}
```

Generator Prompt — image_and_text

Du är en expert på att skapa högkvalitativ träningsdata för informationssökning och Multimodal QA (Question Answering). Den bifogade bilden saknar tillräcklig teckenförklaring och BEHÖVER den omgivande dokumenttexten för att tolkas korrekt.

Utgå från denna text:

```
<text>
{document_context}
</text>
```

REGLER FÖR "BRA" FRÅGOR:

1. Informationsgapet: Texten sätter premissen, men texten FÅR INTE innehålla svaret. Svaret MÅSTE vara dolt i bildens pixlar.
2. Naturlig användarintention: Frågan ska låta som något en riktig användare skulle söka efter.
3. Blind användare: FÖRBJUDET att använda syftningar som

- "enligt kartan", "den röda linjen", "denna" eller "nedan".
4. Explicit svarbarhet: Svaret måste kunna avläsas exakt och otvetydigt från bildens pixlar.

UPPGIFT:

```
{
  "target_insight": "<Beskriv hur texten och bilden kopplas
                    samman>",
  "question": "<Din sökfråga på SVENSKA som följer
              'Blind användare'-regeln>",
  "answer": "<Faktasvaret på SVENSKA, avläst från bilden>",
  "explanation": "<Beviset från bilden på SVENSKA>"
}
```

A.3 VQA Judge Prompt

The following prompt is sent to Qwen2.5-VL-7B-Instruct together with the image and the generated triplet. The system message assigns the model a strict auditor role. The rule string inserted at position {rule_string} differs depending on query type: for `image_and_text` inputs it includes the document context and instructs the judge to accept it as a given premise; for `pure_image` inputs it instructs the judge to treat the pair as fully independent of external context.

System Message

Du är en strikt multimodal domare.
Svara alltid på SVENSKA.

User Prompt

Du är en extremt strikt och skeptisk dataauditor. Granska detta
Fråga-Svar-par kopplat till bilden.

Arbetsgång:

1. Entitetsextraktion: Identifiera alla namn, mätvärden, datum och rumsbeskrivningar i Svar och Förklaring.
2. Oberoende Verifiering: Skanna bildens pixlar själv.
Stämmer entiteterna?
3. Premisskontroll:
{rule_string}
4. Sammanfattande Återkoppling: Vilka entiteter finns i bilden?
Vilka är hallucinerade?
5. Slutgiltigt Betyg (1-5):
 - 1: Allvarlig hallucination.
 - 2: Stora brister.
 - 3: Delvis grundad, småfel.

- 4: Väl grundad, mindre brister.
- 5: Perfekt grundad, 100% korrekt.

UTVÄRDERINGSOBJEKT:

- Fråga: {question}
- Svar: {answer}
- Förklaring: {explanation}

Avsluta EXAKT med: [Betyg X]

Rule String — image_and_text

*** REGEL: KONTEXTUELL PREMISS ***

AI:n fick använda denna bakgrundstext för att formulera frågan:

<text>

{document_context}

</text>

DU MÅSTE ACCEPTERA PREMISSEN. Verifiera att svaret fysiskt finns i bildens pixlar.

Rule String — pure_image

*** REGEL: EXKLUDERAD KONTEXT ***

Detta är en 'pure_image'-fråga. Frågan och svaret måste vara helt oberoende av yttre kontext.

A.4 Text QA Generator Prompt

The following prompt is sent to Mistral Large 3 together with each text chunk. The response format is enforced as JSON with temperature 0.0. The placeholder {chunk_text} is replaced at runtime with the source chunk. If the chunk contains no specific facts suitable for a search query, the model is instructed to return empty fields rather than fabricate content.

User Prompt

Du är expert på att skapa träningsdata för informationssökning.

Givet denna textpassage från ett svenskt kommunalt dokument:

<text>

{chunk_text}

</text>

Skapa EN fråga och ett svar där:

1. Svaret är ett SPECIFIKT faktum (namn, nummer, datum, adress, regel, belopp, gata, tidsperiod)

2. Frågan låter som en naturlig sökning som en kommunalt anställd skulle skriva på svenska
3. FÖRBJUDET: Hänvisa till "texten", "dokumentet", "ovan", "nedan", "denna passage" i frågan
4. Frågan ska vara självständig - en person som INTE sett texten ska förstå vad som söks
5. chunk_excerpt ska vara den EXAKTA ordagranna frasen ur texten som innehåller svaret

Bra exempel:

- "Hur länge får man parkera på Storgatan i Göteborg?"
- "Vad är hastighetsgränsen på väg 40 förbi Landvetter?"
- "Vilket telefonnummer har Bygglovsenheten i Mölndal?"

Dåliga exempel (FÖRBJUDET):

- "Vad säger texten om parkering?" - hänvisar till texten
- "Vilka regler nämns i dokumentet?" - för vagt

Returnera ENDAST giltig JSON:

```
{
  "question": "<naturlig sökning på SVENSKA>",
  "answer": "<specifikt svar från texten>",
  "chunk_excerpt": "<exakt ordagrann fras ur texten som
                  innehåller svaret>"
}
```

Om texten saknar specifika fakta lämpliga för en sökning:

```
{"question": "", "answer": "", "chunk_excerpt": ""}
```

A.5 Text QA Judge Prompt

The following prompt is sent to Qwen2.5-VL-7B-Instruct together with each generated question-answer pair and its source chunk. The judge operates on text only without any image input. The placeholder {excerpt_note} is replaced at runtime with either *"Citatet finns ordagrant i texten."* if the chunk excerpt was found verbatim in the chunk, or *"Citatet hittades INTE ordagrant – kontrollera noggrant."* if it was not.

System Message

Du är en strikt textauditor. Svara på SVENSKA.

User Prompt

Du är en strikt dataauditor för träningsdata inom informationssökning.

Granska detta Fråga-Svar-par och avgöra om svaret är förankrat i den givna texten.

TEXTPASSAGE:

```
<text>
{chunk_text}
</text>
```

FRÅGA: {question}

SVAR: {answer}

PÅSTÅTT CITAT: "{chunk_excerpt}"

CITATVERIFIERING: {excerpt_note}

ARBETSGÅNG:

1. Kontrollera om svaret kan härledas direkt från texten utan extern kunskap.
2. Kontrollera om frågan låter som en naturlig sökning - ingen hänvisning till "texten" eller "dokumentet".
3. Är svaret tillräckligt specifikt (namn, nummer, adress, regel) och inte bara en vag sammanfattning?

BETYGSSKALA:

- 5: Svaret finns ordagrant i texten. Citatet stämmer. Frågan är naturlig.
- 4: Svaret är tydligt förankrat, mindre omformulering acceptabel.
- 3: Svaret är delvis förankrat, viss slutledning krävs.
- 2: Svaret är löst relaterat, stor slutledning krävs.
- 1: Svaret finns INTE i texten / hallucinerat.

Avsluta EXAKT med: [Betyg X]

B

Appendix B: Image Routing Prompts

The following positive and negative prompts are used to route images into seven categories using SigLIP ViT-SO400M embeddings, as described in Section 3.3.1. All prompts are written in English rather than Swedish, since SigLIP was trained predominantly on English image-text pairs and produces more reliable text embeddings for English descriptions in zero-shot classification settings.

Each category is associated with a set of positive prompts, whose cosine similarities to the image embedding are averaged, and a set of negative prompts, whose average cosine similarity is subtracted. The resulting raw scores are Z-score calibrated before routing.

logos_crests — Positive Prompts

A traditional heraldic shield, coat of arms, crest, or crown emblem on a white background.

logos_crests — Negative Prompts

photograph
street map
bar chart
data visualization
paragraphs of text

logos_wordmarks — Positive Prompts

A corporate wordmark, government agency typography logo, or text-based brand identity.

logos_wordmarks — Negative Prompts

photograph
street map
flowchart diagram
bar graph
paragraphs of text

maps — Positive Prompts

An aerial map, satellite view, topographical drawing, or colorful street plan.

A sparse cadastral map, architectural site plan, or zoning map showing grey building footprints, red boundaries, and property lines on a white background.

maps — Negative Prompts

corporate wordmark
heraldic crest
bar chart
flowchart diagram
photograph
paragraphs of text

charts_diagrams — Positive Prompts

A complex data visualization with quantitative X and Y axes.
A statistical bar graph, pie chart, line plot, or scatter plot with data labels.
A business flowchart diagram with connected text boxes and directional arrows.
An architectural schematic, technical blueprint, or engineering cross-section.
A dense tabular data grid, spreadsheet table, or matrix.

charts_diagrams — Negative Prompts

street map
aerial satellite photograph
sparse cadastral property map
a corporate brand logo
a company wordmark or typography
a standalone heraldic crest or shield
a realistic photograph

photos — Positive Prompts

A standard ground-level realistic photograph of an outdoor scene, a building, a landscape, or an indoor room.

photos — Negative Prompts

paragraphs of plain text
bar chart
overhead satellite map view

corporate logo
flowchart

junk_text — Positive Prompts

A scan of plain text, a dense legal document page, or a form with checkboxes.
paragraphs of numbers

junk_text — Negative Prompts

photograph
street map
bar chart
corporate logo
flowchart diagram

junk_stamps — Positive Prompts

A round blue or red ink approval stamp, or a digital signature graphic.

junk_stamps — Negative Prompts

photograph
street map
bar chart
corporate logo

C

Hyperparameters

Tables C.1, C.2, and C.3 list the hyperparameter settings used in all stages of the MAPS training pipeline. Values are taken directly from the implementation and held constant across all experiments unless explicitly varied in the ablation study described in Section 3.7.3. Parameters that appear in multiple stages retain the same value throughout unless a stage-specific override is listed.

Table C.1: Hyperparameter settings for the MAPS training pipeline, general settings and Stages 1–2.

Stage	Parameter	Description	Value
<i>General</i>			
	Gradient clip	Max gradient norm, all stages	1.0
	Weight decay	AdamW weight decay, Stages 1–3	10^{-2}
	Dropout	BridgeBlock dropout rate	0.1
	Expansion factor	BridgeBlock FFN expansion	2
	n (default)	Number of BridgeBlocks	4
	Seed	Global random seed	42
<i>Stage 1: Text-to-Text Pretraining</i>			
	Epochs		5
	Learning rate		10^{-4}
	Batch size		512
	Warmup steps		500
	α	Relational loss weight	0.5
<i>Stage 2: Anchor Fit</i>			
	Epochs	With early stopping	100
	Learning rate		10^{-4}
	Batch size		256
	Warmup steps		200
	τ_{start}	Initial InfoNCE temperature	0.2
	τ_{end}	Final InfoNCE temperature	0.07
	Patience	Early stopping patience	10
	Validation fraction		0.15

Table C.2: Hyperparameter settings for Stage 3 of the MAPS training pipeline.

Stage	Parameter	Description	Value
<i>Stage 3, Phase 1: EWC Anchor</i>			
	Epochs		3
	Learning rate		10^{-5}
	Batch size		256
	Warmup steps		50
	λ	EWC penalty weight	1000
	N_F	Fisher information sample size	500
	Dropout noise	Augmentation noise for consistency	0.3
<i>Stage 3, Phase 2: Context-Grounded Contrastive</i>			
	Epochs		10
	Learning rate		3×10^{-5}
	Batch size		256
	Warmup steps		200
	τ	Soft InfoNCE temperature	0.2
	τ_{gate}	Confidence gate threshold	0.15
	λ_1	Laplacian regulariser weight	0.1
	λ_2	EWC weight carried from Phase 1	1000

Table C.3: Hyperparameter settings for Stage 4 of the MAPS training pipeline.

Stage	Parameter	Description	Value
<i>Stage 4: Unified Multimodal Ranking</i>			
	Epochs	Early stopping; converged at 41	500
	Learning rate (bridge)		3×10^{-6}
	Learning rate (calibration)	For $\log \alpha$ and β	10^{-3}
	Batch size schedule	$B/2$ image + $B/2$ text	$32 \rightarrow 64 \rightarrow 128 \rightarrow 256$
	Batch schedule epochs	Epoch at which each size activates	0, 3, 8, 15
	Weight decay	Stage 4 override	0.05
	τ	Fixed ranking temperature	0.07
	γ	Collapse regulariser weight	0.05
	λ_{disp}	Text displacement penalty weight	0.25
	λ_{α}	L_2 regularisation on $\log \alpha$	10^{-3}
	λ_{β}	L_2 regularisation on β	10^{-3}
	Patience	Early stopping patience	25
	Patience start epoch	Epoch at which patience activates	16
	Validation fraction		0.15
	α at convergence	Learned image score scale	3.57

DEPARTMENT OF PHYSICS
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se



CHALMERS
UNIVERSITY OF TECHNOLOGY