Lost in Compression: The Failure of SSMs to Propagate Factual Context in Transformer Reconstruction

by

Shahriar Khandoker
24141122
Abdullah Al Zubayer
24141076

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2026.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Shahriar Khandoker

24141122

Abdullah Al Zubayer

24141076

Approval

The thesis/project titled "Lost in Compression: The Failure of SSMs to Propagate Factual Context in Transformer Reconstruction" submitted by

1. Shahriar Khandoker (24141122)
2. Abdullah Al Zubayer (24141076)

of Fall, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science in Fall 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Farig Yousuf Sadeque

Associate Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Niloy Farhan

Lecturer
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi

Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

State Space Models (SSMs) have emerged as a promising alternative to Transformer architectures, offering improved computational efficiency and scalability for long-context sequence modeling. Furthermore, several hybrid SSM-Transformer architectures have been proposed, aiming to combine the efficient context modeling capabilities of SSMs with the expressive decoding power of Transformers. These hybrid approaches are commonly claimed to offer advantages over Transformer-only or SSM-only models. In this work, we investigate a critical weakness of such hybrid designs by constructing an encoder-decoder language model in which an SSM-based architecture serves as the encoder and a Transformer serves as the decoder. Although this model produces grammatically fluent outputs, we observe systematic degradation in factual accuracy when evaluating it on text summarization tasks. Through controlled experiments, we demonstrate that the latent representations generated by the SSM encoder are insufficient for accurate factual reconstruction during Transformer-based decoding. Our findings suggest that SSM-based encoders compress contextual information in a manner that is not fully decodable for factual and faithful summarization, exposing a fundamental limitation of SSMs in encoding context-rich representations suitable for downstream reconstruction. This work provides insights into the challenges of fact-preserving context compression in neural language models.

Keywords: State Space Models (SSMs), Mamba, Hydra, Hybrid SSM-Transformer Architecture, Encoder-Decoder Model, Long-Context Modeling, Context Compression, Abstractive Summarization, Hallucination, Factual Consistency, Faithfulness, Latent Representation, Bridge Adapter.

Acknowledgement

We would like to express our sincere gratitude to our supervisor, Dr. Farig Yousuf Sadeque Sir and our co-supervisor, Niloy Farhan Sir for their guidance and support throughout this thesis. We are also thankful to our friends and family for their constant encouragement.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI Artificial Intelligence

BERT Bidirectional Encoder Representations from Transformers

BPE Byte-Pair Encoding

C4 Colossal Clean Crawled Corpus

GLUE General Language Understanding Evaluation

GPT Generative Pre-trained Transformer

GPU Graphics Processing Unit

KD Knowledge Distillation

LLM Large Language Model

LM Language Model

MDS Multi-Document Summarization

MLP Multi-Layer Perceptron

NLP Natural Language Processing

OOV Out-Of-Vocabulary

PPL Perplexity

Q, K, V Query, Key, and Value

RNN Recurrent Neural Network

RoPE Rotary Positional Embedding

ROUGE Recall-Oriented Understudy for Gisting Evaluation

RWKV Receptance Weighted Key Value

S4 Structured State Space for Sequences

S6 Selective State Space Model

SOTA State-of-the-Art

SSD State Space Duality

SSM Structured State Space Model

T5 Text-to-Text Transfer Transformer

TL;DR Too Long; Didn't Read

XAI Explainable Artificial Intelligence

Chapter 1

Introduction

1.1 Background

For more than a decade, sequence modeling in natural language processing (NLP) encompassing tasks such as language modeling, machine translation, question answering, and abstractive text summarization-has been dominated by architectures based on the Transformer and its self-attention mechanism [74]. The central strength of Transformers lies in their ability to model arbitrary long-range dependencies via dense pairwise interactions between all tokens in a sequence. This architectural property enables strong contextual representations and has driven state-of-the-art performance across a wide variety of benchmarks [74] [71].

Despite their success, vanilla Transformers have a computational drawback the quadratic time and memory complexity of self-attention as a function of sequence length [65]. With increasing input sequence length, the price of computing and storing attention matrices is prohibitive. This makes the Transformer-based systems highly limited in terms of scaling to long-context tasks, including full-document summarization [57], legal and financial document analysis, long-form dialogue modeling, and scientific literature synthesis. In these fields, the pertinent information may be distributed sparsely across tens or hundreds of thousands of tokens, but the model cannot be deprived of the ability to access fine-grained factual information along the entire sequence.

To mitigate this quadratic bottleneck, a wide range of architectural modifications have been proposed. Sparse-attention models such as Longformer [32] and BigBird [49] reduce complexity by restricting attention to local or block-sparse patterns, while recurrent extensions such as Transformer-XL [21] introduce segment-level recurrence to propagate information across chunks. Later on, several sub-quadratic and linear-time sequence model families have been introduced, such as Performer [38], Hyena [70], RetNet [72], S4 [64], Mamba [81], and related State Space Models (SSMs) [92]. These architectures either substitute or estimate self-attention with convolutional, kernel-based or state-space dynamics, which enable sequences to be run in linear time with respect to length.

Among these alternatives, SSM-based architectures have been of particular interest as they have good scaling behaviour and favourable inductive biases to long-range

sequence processing. [64] [81]. Its ability to store information with only a compact latent state which recurrently changes over time means that SSMs are able in practice to integrate information in arbitrary length contexts and requires only linear memory and compute. Recent variants such as S4 [64] and Mamba [81] have competitive performance with Transformers on a variety of long-sequence benchmarks [88] [75], and much lower inference and training cost.

Although, there exist these computational advantages, there is a trade-off. In contrast to Transformers, token-level representations are explicitly maintained by dense attention maps [74], SSMs necessarily compress the input sequence into a fixed-dimensional latent trajectory. The result of this compression process leads to one crucial and understudied question: to what extent can factual and entity-level information be faithfully preserved and later when using such compressed representations, especially when the downstream task requires precise recall or attribution of facts from distant parts of the input?

The question is particularly relevant in the situation of abstractive text summarization [27] [50] [28]. Summarizing is not just a language generation problem, it is also a factual reconstruction problem. The model has to find the salient facts scattered in a long document, store them in its internal representation, and generate a concise summary that remains faithful to the source [26] [45]. Any loss of the factual context is likely to result in the generation of fluent and yet factually wrong or hallucinated summaries.

In order to balance SSMs' efficiency and the expressive capability of Transformers, a growing body of work has proposed hybrid architectures [84] [91] [80] [78] that combine SSM and Transformers. In these models, an SSM-based encoder processes the long input sequence efficiently, while a Transformer-based decoder generates the output sequence using attention over the encoded representations. These hybrid methods are said to be the best in the world: they provide the scalability of long-context encoding combined with high-quality natural language generation.

Regardless of these claims, the representational sufficiency of SSM-based encoders of downstream factual reconstruction has not been rigorously interrogated. In particular, it remains unclear whether the latent states that SSM encoders output contain enough fine-grained factual information to be faithfully decoded by a Transformer. This thesis posits itself at exactly this unresolved junction.

1.2 Motivation

The motivation for this research arises from an increasing tension between two opposing desiderata in modern NLP systems: scalability to long contexts and faithfulness of generated content. Practical uses on the one hand are more and more requiring models with the ability to handle very long documents, such as multi-chapter reports, legal filings, or collections of scientific papers [67], [75]. On the other hand, the same applications are very strict in terms of factual accuracy, traceability and faithfulness to source material [26].

Hybrid SSM-Transformer architectures appear, at least superficially, to provide a graceful resolution to this conflict. A common design uses SSM-based encoder to compress a long input document into a sequence of latent vectors by using linear-time recurrent dynamics [81]. A Transformer decoder then conditions on these latent vectors to generate an abstractive summary [27]. The assumption behind this is the belief that the SSM encoder will be able to compress the raw content of the document into a form that will be fully decodable by the Transformer.

This assumption is, however, no trifle. This is not just an alternate implementation of attention, SSM encoders introduce a different inductive bias [64]. Instead of explicitly attending to all pairs of tokens, they propagate information through a low-dimensional hidden state governed by linear or nonlinear dynamics. This mechanism inevitably performs a form of information bottleneck or compression.

From an information-theoretic perspective, any compression that reduces a high-dimensional sequence into a lower-dimensional latent trajectory must discard some information. The critical question is not whether information is lost, but which information is lost. When syntactic regularities and coarse topical signals are preserved while fine-grained factual associations are degraded, then the output of the resulting model can be fluent and coherent but systematically unfaithful [26] [45].

This is exactly the failure mode that has been alluded to by recent anecdotal and empirical evidence. A number of studies indicate that the SSM-based models exhibit weaker copy behavior, lower entity fidelity, and degraded performance on tasks requiring exact recall [88]. Nevertheless, these findings have not been systematically discussed in terms of hybrid encoder-decoder designs, nor have they been framed as a fundamental limitation of context compression.

The rationale behind this thesis is that the hypothesis that SSM-based encoders are computationally appealing but compress the contextual information in a manner that is not fully decodable for factual reconstruction. In other words, even in spite of a powerful Transformer decoder, the information bottleneck introduced by the SSM encoder may irreversibly erase or entangle factual details that is required for faithful summarization.

Rather than assuming that hybridization is inherently beneficial, this work adopts a more critical stance. Instead of viewing the hybrid SSM-Transformer architecture as a proven improvement over pure Transformers, it is viewed as a testbed to explore what neural sequence models are capable of doing in the name of compressing facts.

1.3 Problem Statement

The central problem addressed in this thesis is the following:

Although hybrid SSM-Transformer architectures [84] [91] [80] are widely proposed as efficient alternatives to full-attention encoder-decoder models, it is not known

whether the latent representations produced by SSM-based encoders are sufficient to support faithful factual reconstruction during Transformer-based decoding.

More concretely, this research investigates whether an SSM encoder can preserve the fine-grained factual and entity-level information required for abstractive summarization, or whether the compression inherent to state-space dynamics introduces irreversible information loss that manifests as hallucinations, omissions, or factual distortions in generated summaries.

We focus on a specific instantiation of this problem by constructing an encoder-decoder language model in which:

- The encoder is implemented using a modern SSM-based architecture (e.g., Hydra: Mamba-style quasi-separable mixers) [83].
- The decoder is borrowed from a T5 [71] Transformer decoder [74] with cross-attention over encoder outputs.

Despite producing grammatically fluent outputs, such a model may fail in a more subtle and insidious way: by systematically degrading factual accuracy relative to Transformer-only baselines. This failure mode cannot be detected by surface-level fluency metrics alone [3] and requires targeted factuality evaluation [26] [45].

The core research question can therefore be stated as:

Do SSM-based encoders preserve sufficient factual context to enable faithful reconstruction by a Transformer decoder, or do they impose a lossy compression that fundamentally limits downstream factual accuracy?

1.4 Objective

The primary objective of this research is not to design a better summarization model per se, but to diagnose and characterize a failure mode of hybrid SSM-Transformer architectures.

Specifically, the objectives are as follows:

- To design and implement an encoder-decoder language model in which a modern SSM-based architecture [81], [83] serves as the encoder and a Transformer serves as the decoder.
- To construct controlled experimental settings in which the hybrid model can be directly compared against Transformer-only encoder-decoder baselines [27], [50] on abstractive summarization tasks.
- To evaluate not only surface-level generation quality (e.g., ROUGE [3], BERT-Score [51]), but also factual faithfulness using factuality-sensitive metrics [26], [45] and human or QA-based evaluation protocols.

- To empirically test the hypothesis that SSM encoders introduce a lossy compression of contextual information that degrades factual reconstruction during decoding.
- To analyze the latent representations produced by the SSM encoder and identify qualitative and quantitative signatures of information loss, such as weakened entity retention, reduced copy behavior, and degraded long-range factual attribution.
- To demonstrate that grammatical fluency is not a sufficient indicator of representational adequacy, and that hybrid models can appear superficially successful while failing at factual preservation.
- To present the broader implications of these findings for the design of long-context language models and for the feasibility of fact-preserving context compression.

1.5 Methodology in Brief

This research adopts an empirical and analytical methodology that focuses on the design and evaluation of hybrid encoder-decoder architectures.

The encoder part is implemented using a quasi-separable matrix mixer architecture derived from modern SSM designs, named Hydra [83]. This model operate in linear time in a recurrent state dynamics with low-rank parameterizations over the input sequence. The encoder generates a series of latent vectors that are supposed to be summarizing the input document.

The decoder part is developed as a standard Transformer decoder with masked self-attention and cross-attention over the encoder outputs. This structure resembles the traditional sequence-to-sequence models such as T5 [71] and PEGASUS [50], allowing for controlled comparisons.

A projection layer is added to create a bridge between the representational mismatch between the SSM encoder and the Transformer decoder. This layer maps the latent states of the encoder into the dimensionality and distribution of the cross-attention mechanism of the decoder. This interface is considered as a potential bottleneck and is analyzed closely.

The hybrid model is trained or fine-tuned on standard abstractive summarization datasets such as CNN/DailyMail [9] and XSum [20]. For each dataset, matched Transformer-only encoder-decoder baselines are trained under comparable conditions.

It is evaluated with regard to the following three axes:

- Surface-level quality: ROUGE [3], BERTScore [51].
- Factual faithfulness: QA-based factuality metrics [45], hallucination rate, entity consistency [26].

- Efficiency: inference speed, memory usage, and scalability with sequence length.

More importantly, the primary emphasis of this thesis is not on efficiency gains, rather on the diagnosis of representational failure. Measurements of efficiency are considered as secondary context.

1.6 Scopes and Challenges

This research is defined by specific boundaries and predetermines some key challenges that will need to be addressed.

Scopes:

- The study is restricted to encoder-decoder language models for abstractive summarization [27] [50] [28].
- The linguistic domain is limited to English-language text.
- Model sizes are constrained to the base range (approximately 100M-300M parameters).
- The focus is on architectural behavior and representational adequacy, not on achieving state-of-the-art benchmark scores.

Challenges:

- Fundamental representational mismatch between recurrent SSM encoders and attention-based decoders [74] [81].
- Difficulty of disentangling whether factual degradation arises from the encoder, the interface layer, or the decoder's inability to utilize compressed representations.
- Engineering challenges in integrating custom SSM kernels into standard sequence-to-sequence training pipelines.
- Potential irreversibility of information loss introduced by SSM compression.

Chapter 2

Literature Review

2.1 Preliminaries

2.1.1 Transformer Architecture and Self-Attention

The Transformer architecture is the de facto standard for high-quality and SOTA sequence modeling [74] [82]. Its central mechanism, the self-attention module, is the key innovation that allows the model to dynamically weight the importance of each token in the sequence with all other tokens when producing a relationship between tokens. This mechanism is constructed through three learnable vector representations for each input token: Query(Q), a Key(K), and a Value(v). The Query represents the current token's request for information, the Keys represent the information that other tokens have to offer, and the Values represent the content of those other tokens.

The core computation is the scaled dot-product attention, which is mathematically described as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Here, the dot product QK^T computes a matrix of similarity scores between every Query and every Key. In order to stabilize the gradients, the results is scaled by the square root of the key dimension, $\sqrt{d_k}$. These results or logits are then converted into a probability distribution or attention with the help of softmax function. These distributions are then used to compute the sum of Value vectors,
The important aspect of this mechanism is QK^T matrix multiplication. This computes a score for every token to every other token. If length is N, then its computation and memory complexity is $O(N^2)$ [65]. This quadratic scaling is the fundamental source of Transformers models' inefficiency on long sequences. This problem is what inspired us to solve through our research

2.1.2 Subquadratic Models

The quadratic complexity of the self-attention mechanism is a fundamental bottleneck of Transformers, which imposes a substantial obstacle to efficient processing of

very long sequences.The quadratic complexity of the self-attention mechanism is a fundamental bottleneck of Transformers, and poses a significant barrier to efficient processing of very long sequences [85]. This limitation has prompted a wide-scale research to develop subquadratic model architectures to process input sequence with time and memory complexity better than the $O(n^2)$ scaling of traditional self-attention mechanisms of Transformers where n is the sequence length. These models offer competitive, or even superior performance in tasks that involve long contexts while realizing substantial improvements in computational and memory efficiency, which is imperative for the ability to scale foundation models to longer inputs as well as more resource-constrained environments.

Subquadratic models can be roughly grouped into 2 categories: techniques for approximating part (a) the full attention mechanism and (b) completely new architectural paradigms.

Attention approximation aims to lower the calculation cost while maintaining the major structure of the Transformer.

Sparse Attention: Some models such as Swin Transformer and BigBird add sparsity to the attention matrix [85]. Instead of having all tokens attend to all other tokens, attention is limited to a smaller, well-defined subset of tokens, e.g. local windows or global tokens. While this leads to a reduction in complexity, it might also be the main limitation to the models capacity to understand arbitrary, long-range dependencies (that full attention is great at learning) [86].

Linear-Attention: Those models are reparameterizing or approximating the attention mechanism to allow linear time operations. It is a replacement of the softmax computation, which is expensive, by kernel-based approximations. This is just a mathematical reformulation, and there is a possibility of changing the order of the matrices so that the final complexity became not quadratic, but linear, and more precisely, O(n) time and space complexity. Similarly, models such as Linformer [46] and Longformer [32] limit attention range to fixed-size attention windows or low-rank projections, severely limiting the overhead in long sequences.

The second category is that of a design of alternate architectures that are subquadratic, i.e. inherently are subquadratic. Moving away from the attention mechanism altogether, they have different principles of modeling sequences.

Structured State Space Models (SSMs): As will be detailed in the following section, SSMs completely avoid attention, instead viewing sequence modeling in terms of a dynamic system, i.e. a *state space model* [92]. By leveraging the structure of linear recurrences and efficient convolution (either through FFTs or gating mechanisms), these models achieve linear scaling in the sequence length inherently, with global context sensitivity and performance that is competitive with Transformers, so they become the focus of this thesis.

Hybrid Recurrent Models: Models such as RWKV [69] are a hybrid approach between RNN-like recurrence and a Transformer-like expressiveness. The key inno-

vation, the Receptance Weighted Key Value modulates input representations using gating mechanism inspired by attention - while keeping causal recurrence. Unlike classical RNNs, they are designed to be parallelizable in the training process like Transformers, but efficient in the inference process like RNNs and provide another avenue towards sub-quadratic scaling.

2.1.3 Structured State Space Models / SSMs

As an alternative to Transformers [74] that have been able to address the weakness of the self attention quadratic time and compute complexity [65] , State Space Models were introduced [64]. Mamba-2 [77] is the current SOTA amongst SSMs with superior performance to Transformer models of comparable size and introducing many improvements [88] [92].

SSMs are based on classical control theory. They approximate the behavior of a system in terms of a latent state that developed over time [64]. A continuous-time, linear time-invariant SSM is defined by two core equations:

$$x'(t) = Ax(t) + Bu(t) \tag{2.1}$$

$$y(t) = Cx(t) + Du(t) \tag{2.2}$$

In this formulation, u(t) is a 1D input function (e.g., a single token embedding over time), x(t) is an N-D latent state vector, which represents a compressed representation of the history of the system, and y(t) is the 1D output. The matrices A,B,C, and D are parameters that control the dynamics of the system [64].

To use this continuous system in a deep-learning model, we first pick a time step Δ and apply a discretization rule (for example, the bilinear transform). This turns our continuous matrices (A,B) into discrete ones ($\bar{\text{A}}$,$\bar{\text{B}}$) [64] [81]. This discretization gives SSMs two equivalent views:

Recurrent View
We get a simple linear recurrence:

$$x_k = \bar{A}x_{k-1} + \bar{B}u_k, \quad y_k = \bar{C}x_k.$$

Here, the state x_k always has size N, no matter how long the sequence is. That means each step takes the same small amount of compute and memory, so inference is fast and efficient [81].

Convolutional View
If you unroll the recurrence, the whole output sequence y is just the input u convolved with a kernel $\bar{K}$ built from $\bar{A}, \bar{B}, \bar{C}$:

$$y = \bar{K} * u.$$

This form can be computed in parallel across the sequence-perfect for GPUs and other accelerators.

By switching between these two views, SSMs let us train quickly in parallel and still run inference step-by-step efficiently, all with a cost that grows only linearly in the sequence length.

2.1.4 Mamba Architecture: Selective State Space Models (S6)

The Mamba [81] architecture is a major development of the SSM architecture, which overcomes an important shortcoming of its predecessors such as S4 [64]. While traditional SSMs work with fixed and time invariant descriptors for system matrices (A, B, C), Mamba introduces a selection mechanism which renders these system parameters input-dependent and time-varying.

The main innovation that the Mamba introduces is its selective scan (S6) [81] in which the key parameters, namely the input projection matrix B, the output projection matrix C and the discretization step size Δ are no longer fixed. Instead, they are dynamically generated from the input token x_t at each timestep. typically through simple linear projections. This seemingly trivial change has had massive improvement in performance [81] [88]. By making the system matrices functions of the input, Mamba can modulate its behavior token by token basis.

If a token is important, the model can generate a large B vector to ensure its information is fully integrated into the state x_t. But if the current context is no longer relevant then the model can generate a specific Δ that effectively resets or dampens the state, allowing it to forget irrelevant history [81].

This content-aware selection mechanism gives Mamba the ability to handle complex and context-rich information-dense semantic meaning of natural languages, a domain where prior SSMs struggled to compete with Transformers [81] [88]. In practice, this mechanism is essentially very similar to self-attention and its core benefits, allowing it to compete and sometimes surpass the performance of comparable Transformer models [88]. Thus, Mamba has been propelled to achieve SOTA performance, establishing it as the first subquadratic architecture to truly match Transformer-level quality on major language modeling benchmarks [81] [92].

2.1.5 Hydra: Generalized Matrix Mixers and Bidirectionality

Hydra [83] is based on Mamba architecture. It implements the inclusion of a quasiseparable matrix mixer operation. Hwang et al. proposes viewing all sequence models as applying a mixing matrix M to the input X [83].

- Attention: M is a dense, data-dependent matrix.

- Causal SSM: M is a lower-triangular semiseparable matrix.

A matrix is quasiseparable if its off-diagonal blocks have low rank [83]. This structure is profound because it allows for bidirectional recurrence. While Mamba is inherently causal (Forward only), Hydra can process information in both directions (Forward and Backward) effectively within a single linear-algebraic framework, rather than simply concatenating two independent models.

An encoder has to be bidirectional to fully grasp the context of a token based on its surroundings (both past and future). A standard Mamba layer has a causal. Using Hydra as the encoder gives access to the required global context visibility of a BERT-like model, but with the $O(L)$ complexity of an SSM [83]. This is the corner-stone of the feasibility argument for this thesis.

2.2 Review of Existing Research

The rapid evolution of efficient sequence modeling has spurred a diverse body of research exploring the intersection of State Space Models (SSMs) and Transformers [92]. This review categorizes existing literature into three primary streams: foundational bidirectional SSMs, hybrid architectural paradigms, and SSM-based sequence-to-sequence applications.

2.2.1 Evolution of Bidirectional SSMs

While the original Mamba architecture [81] proved that selective SSMs can be as good as Transformer for causal language modeling, its natural unidirectionality is quite a drawback for non-causal tasks like document encoding or image processing. Early attempts to solve this issue, such as Vision Mamba (Vim) [89] adopted "Bi-Mamba" approach where two independent SSMs (forward and backward) are run and concatenated. This approach, although effective, doubles the number of parameters and memory of the sequence mixing layers.

Hydra [83] is a more important development in this field. By generalizing Mamba using the framework of quasiseparable matrix mixers, Hydra achieves bidirectionality through sharing of core projection weights across directions. This enables it to model world context with the same $O(L)$ linear complexity as Mamba but without the parameter inefficiency of previous bidirectional heuristics.

2.2.2 Hybrid Architectures

Recognizing the complementary strength of SSMs (efficient long-context retention), Transformers (precise retrieval and in-context learning), in recent literature focus has been increasing on hybrid architectures.

- **Interleaved Hybrids:** The Jamba [84] architecture shows the scalability of mixing layers, interleaving Mamba and Transformer blocks at a ratio of 7:1, and at a massive 52B parameter scale. This design uses attention layers as synchronization points" to recover fine-grained details which are too much

compressed by pure SSMs. Similarly, Samba [91] alternates layers of Mamba with Sliding Window Attention (SWA) to provide state-of-the-art performance on infinite-context benchmarks.

- **Parallel and Head-Level Hybrids:** Hymba [78] proposes to hybridize the layer itself, as it substitutes particular attention heads with SSM heads to trade-off fine-grained recall and efficient summarization. Zamba [80]uses a primarily SSM backbone with a single shared global attention module; this greatly decreases the memory overhead of Key-Value (KV) caches.

- **Positional Consistency in Hybrids:** The key problem with hybridization is making the implicit positional awareness of SSMs consistent with the explicit positional demands of Transformers. TransXSSM [93] addresses this by coining a Unified Rotary Positional Embedding (RoPE) [60] which makes both modules types coherent in position encoding.

2.2.3 Sequence-to-Sequence and Encoder-Decoder Models

Research that specifically applies SSMs to the encoder-decoder framework-the standard for summarization-remains more limited but provides essential baselines.

- **Pure SSM Seq2Seq:** LOCOST [76] proposes a pure SSM encoder-decoder model to summarize long-document content with long-state space models. Although it creates dramatic decreases in memory (up to 87% inference memory reduction versus sparse Transformers), it also performs slightly worse in ROUGE evaluations, indicating that a pure SSM decoder is not as fidel to the generation processes as attention-based mechanisms are.

- **Inverse Hybrids:** TransMamba [94] will go to the hybrid encoder-decoder but instead follows the inverse arrangement of the proposed thesis: it encodes with a Transformer but decodes with a Mamba. Although it works well on the tasks which involve complex relational thought in the encoder, this design fails to address the quadratic bottleneck of encoding long documents, the main limitation of summarization.

2.3 Summary of Key Findings

The literature synthesis has shown that there is a clear trend toward hybrid models as the best approach to efficiency-expressivity tradeoff, and also that there is a certain gap in encoder-decoder design of summarization. The major results that led to this thesis are:

1. **Theoretical Compatibility (State Space Duality):** Recent theoretical work on State Space Duality (SSD) theoretically shows that the recurrent states of modern SSMs like Mamba-2 are mathematically equivalent to a restricted form of linear attention. This implies that the hidden states produced by an SSM encoder are structurally compatible with the Key-Value interfaces expected by a Transformer decoder, validating the feasibility of the proposed hybrid architecture.

2. **The "Encoder Gap" in Hybridization:** Most existing hybrids (Jamba, Zamba, Samba) are decoder-only language models designed for generative tasks. The few encoder-decoder hybrids, such as TransMamba, prioritize Transformer encoders. There is a lack of research on Hydra-Encoder/Transformer-Decoder architectures, specifically designed to exploit the asymmetry of summarization: utilizing a linear-time SSM to compress massive inputs ($L \gg 10k$) while retaining a quadratic-time Transformer for high-quality generation of short outputs.

3. **Necessity of Quasiseparable Bidirectionality:** The standard causal SSMs would never be effective encoders since they would not be seen in the future. Although there are bidirectional versions such as BiMamba, they pay a factor-2 parameter tax. It has been found that the quasiseparable matrix approach of Hydra can solve this by allowing bi-directional context mixing without parameter doubling, and is therefore the most efficient candidate to replace a drop-in encoder.

4. **Efficiency vs. Recall Trade-off:** According to evidence provided by LOCOST and Jamba, pure SSMs are efficient, but do not appear to tackle the needle-in-a-haystack recall tasks particularly well as Transformers. A more theoretically well-grounded way to reduce this trade-off is to use a hybrid that leaves the bulk of context compression to a Hydra encoder, but then lets a Transformer decoder cross-attend to the states thus compressed.

In conclusion, the literature substantiates the personal effectiveness of Hydra and Transformers but show us an opportunity that has not been investigated yet in overcoming the quadratic bottleneck of long-document summarization through a hybrid.

2.4 Summary of Existing Hybrid Models

2.4.1 TransMamba: Transformer-Mamba Encoder-Decoder Hybrid

TransMamba [94] proposes a hybrid encoder-decoder architecture in which a Transformer-based encoder feeds into a Mamba-based decoder, with a learned feature-fusion mechanism mediating cross-attention between the two components. The Transformer encoder is responsible for extracting rich relational representations, while the Mamba decoder leverages state-space dynamics for efficient sequence generation. Experiments on a range of language understanding tasks, including commonsense and science question answering, show that TransMamba outperforms both Transformer-only and Mamba-only baselines. However, the architecture is relatively complex and requires careful feature fusion. Importantly, TransMamba represents the inverse design of SSM-as-encoder approaches: SSMs are used for decoding rather than encoding. As such, its results do not directly address whether SSM-compressed representations are decodable by Transformers for tasks such as summarization, nor does it report long-context or summarization-specific evaluations.

2.4.2 Hymba: Hybrid-Head Transformer-SSM Models

Hymba [78] introduces a head-level hybrid architecture in which standard Transformer layers are augmented by replacing a subset of attention heads with Mamba-based SSM heads. This design allows attention heads to preserve fine-grained token-level recall while SSM heads efficiently summarize longer-range context. Evaluated on a wide range of reasoning benchmarks, Hymba models under 2B parameters outperform similarly sized Transformer-based models and even exceed larger baselines such as LLaMA-3 in some settings. The approach demonstrates strong throughput and efficiency at sequence lengths up to 8K tokens. Nevertheless, Hymba is not mainly judged as a sequence to sequence or a summarization framework, but rather as an autoregressive system. Although its hybrid heads hint that SSM-based context compression may co-exist with attention mechanisms, compatibility of encoders and decoders and recovery of facts out of compressed representations are not directly considered within the model.

2.4.3 Jamba: Transformer-Mamba Mixture-of-Experts Language Model

Jamba [84] is an extensive hybrid language model that injects Transformer and Mamba layers into a Mixture-of-Experts (MoE) setting, allowing it to scale freely to long contexts with competitive language modeling performance on common language modeling benchmarks. The model can support contexts of up to 256K tokens and has good throughput in spite of the size. Jamba gives strong arguments that SSM-Transformer hybrids may be useful at very long contexts in autoregressive environments. Its analysis however is more on perplexity and in-context learning and not so much on structured generation learning like summarization. Additionally, the implication of MoE routing and mixed-layer stacks is architectural complexity that is quite different to encoder-decoder configurations. Consequently, Jamba does not explicitly answer the question of whether, with SSM-based encoders, it is possible to generate representations which can be faithfully decodable by Transformer-based decoders.

2.4.4 Samba: Alternating SSM and Sliding-Window Attention

Samba [91] proposes a hybrid architecture that alternates Mamba-based state-space layers with sliding-window self-attention layers, combining global context compression with localized attention. Trained at scale, Samba achieves state-of-the-art perplexity on long-context language modeling benchmarks and demonstrates strong retrieval performance at sequence lengths up to one million tokens. The model achieves near-linear scaling with context length and significant throughput improvements over full-attention Transformers. Samba's design suggests that SSMs are effective at retaining global contextual information, particularly when complemented by local attention. While this architecture is promising for long-input tasks such as summarization, it has not been evaluated in encoder-decoder settings, nor does it analyze the fidelity of reconstructing factual content from SSM-compressed representations.

2.4.5 Zamba: Mamba Backbone with Global Shared Attention

Zamba [80] presents a relatively small hybrid design that makes use of a Mamba SSM backbone extended with one global shared self-attention (GSA) module. The shared attention mechanism significantly decreases memory overhead, as one set of key-value pairs is shared between layers, allowing their inference to be done effectively without a standard attention cache. Zamba outperforms by a significant margin even much larger Transformer models on reasoning benchmarks at significantly reduced efficiency costs. Nevertheless, Zamba is described more as a decoder-only language model and the global attention is not directly applied to encoder-decoder cross-attention. Though the architecture implies that restricted attention can be used in complement to SSM-based context modeling, it does not assess whether the representations can be reliably inferred in generating facts.

2.4.6 Zamba2: Refined Mamba2-Transformer Hybrids

The Zamba2 family [79] extends the original Zamba architecture by incorporating Mamba-2 (selective SSMs) alongside minimal Transformer components, resulting in improved efficiency and performance across multiple model scales. Trained on the large-scale Zyda-2 corpus, Zamba2 models achieve state-of-the-art perplexity among open-weight language models of comparable size. These results highlight the effectiveness of refined SSM architectures when combined with limited attention. Nevertheless, Zamba2 models are evaluated primarily as autoregressive language models, and their applicability to encoder-decoder summarization remains unexplored. In particular, the literature does not examine whether SSM-based compression in these hybrids preserves the fine-grained factual information required for faithful reconstruction by a Transformer decoder.

2.5 Unexplored Encoder-Decoder Compatibility in SSM-Transformer Hybrids

The analysis in 2.4 of existing hybrid architectures combining State Space Models (SSMs) and Transformers reveals that the majority of proposed models follow one of two architectural paradigms. The first interleaves SSM layers with Transformer attention layers within a single backbone, while the second employs a Transformer-based encoder paired with an SSM-based decoder. Representative examples of the former include models such as Jamba [84], Zamba [80], and Samba [91], whereas TransMamba [90] [94] exemplifies the latter design. Although these approaches incorporate SSMs to varying degrees, they share a common reliance on quadratic self-attention for core language understanding and generation.

As a result, the efficiency and long-context benefits offered by these hybrid models remain incremental rather than transformative. In interleaved architectures like Samba, attention mechanism continues to determine computational cost and representational capacity, with SSM components primarily serving as secondary helpers layers. Similarly, when SSMs are used as decoders, language generation remains

autoregressive and sequential, limiting potential efficiency gains. In both cases, the most computationally expensive operation-autoregressive token-level attention-remains unavoidable, and SSMs are not leveraged to fundamentally alter the language modeling pipeline.

This design choice overlooks the most promising theoretical advantage of SSMs: their ability to scan and compress long sequences efficiently through linear-time state evolution [77], [81]. If attention mechanisms are already responsible for autoregressively scanning and encoding context during generation, then inserting SSM layers offers limited additional benefit. Consequently, existing hybrid models do not fully exploit SSMs as primary context encoders, and instead use them in roles where their strengths overlap with, rather than replace, attention-based processing.

A more consequential efficiency gain would arise from reversing this paradigm-using SSMs exclusively for long-context scanning and encoding, while delegating language generation to Transformer decoders. Such an encoder-decoder separation would allow most of the costly autoregressive computation to be eliminated during encoding, preserving attention only where it is most effective: structured decoding and conditional generation. However, this architectural direction remains largely unexplored.

As a result, a critical gap remains in the research. It is currently unclear whether Transformer decoders can faithfully generate language from the compressed latent representations produced by SSM-based encoders. This gap is important for summarization and other fact-sensitive generation tasks where failures in representation decodability may generate texts as fluent but factually incorrect outputs. The present work directly addresses this unresolved question by constructing an SSM-encoder/Transformer-decoder architecture and evaluating the factual decodability of SSM-compressed latent matrices.

Chapter 3

Specifications, Requirements, and Project Impact

This chapter formalizes the technical scope, evaluation criteria, and wider implications of the research described in this thesis. Section 3.1 outlines the final system specifications and research requirements which operationalize the study of factual degradation in hybrid SSM -Transformer architectures. Section 3.2 discusses how a factual failure in neural summarization systems affects society, particularly in high-stakes informational contexts. Section 3.3 presents the environmental motivations that led to the investigation of SSM-based encoders, and critically analyzes the trade-off between computational performance and information fidelity.

3.1 Final Specifications and Requirements

This section defines the goals, design specifications, and evaluation requirements that will regulate the construction and analysis of the hybrid SSM-Transformer encoder-decoder model applied in this study.

Instead of putting Structured State Space Models (SSMs) in direct competition with Transformer encoders, this study directly evaluates their limitations as contextual compressors in reconstruction-based generation tasks. The core hypothesis is that SSM-based encoders, even with their preferred computational properties, compress contextual information in a way that is not completely decoded by Transformer-based decoders and results in systematic factual degradation. This experimental framework is not thus meant to optimize the performance of peak summarization, but to reveal and describe failure modes of factual propagation.

Implementation of all system components is done based on reproducible research requirements, modular design principles and dataset governance requirements. The experiments are guided by ethical and licensing considerations of benchmark datasets like XSum [20] and CNN/DailyMail [9], ensuring responsible use and comparability with previous work.

3.1.1 Objectives

The main objective of this thesis is to diagnose and empirically validate a failure mode in hybrid SSM-Transformer architectures, in which the latent representations generated by an SSM encoder are not sufficient for faithful factual reconstruction by a Transformer decoder.

The following are the specific objectives:

Architectural Objective: To construct an encoder-decoder language model in which a Structured State Space Model (specifically a Mamba-based encoder variant called Hydra) functions as the encoder, while a Transformer-based architecture serves as the decoder, enabling controlled study of cross-architectural information transfer.

Representational Objective: To evaluate whether the latent states generated by an SSM encoder preserve factual relations, entities, and long-range dependencies in a form that is recoverable by an expressive decoder, independent of surface-level fluency.

Experimental Objective: To assess the model on abstractive summarization benchmarks using both lexical overlap metrics (e.g., ROUGE [3]) and factual consistency metrics (e.g., BERTScore [51], QAEval [61] , FactCC [52]), explicitly separating linguistic fluency from factual faithfulness.

Analytical Objective: To analyze how compression characteristics of SSMs-such as sequential state updates and implicit memory decay-affect the fidelity of reconstructed content, particularly under increasing input length and information density.

Research Contribution Objective: To provide empirical evidence that challenges the assumption that efficient sequence modeling necessarily yields decodable context representations, thereby contributing to a more nuanced understanding of SSM-Transformer hybrid designs.

3.1.2 Functional Requirements:

The functional requirements determine the minimal operational capabilities required to support the research hypotheses.

FR-1: Text Input Processing: The system is required to ingest tokenized documents written in the English language out of standardized summarization datasets (primarily XSum), supporting variable-length inputs where truncation does not cause the loss of information.

FR-2: Abstractive Summary Generation The model needs to generate fluent abstractive summaries based on the latent representations of the encoder so that the quality of reconstruction can be evaluated instead of extractive copying.

FR-3: Implementation of SSM Encoder: The encoder should be designed as a Structured State Space Model architecture (Hydra/Mamba variant), preserving its sequential state update dynamics without auxiliary attention mechanisms.

FR-4: Transformer Decoder Implementation: The decoder should be a standard Transformer-based autoregressive model compatible with a WordPiece tokenizer, with enough expressive power to recreate factual information if such information is present in the latent representation.

FR-5: Adaptation Layer of Latent Space: An intermediate adapter layer must project the SSM encoder's output into the decoder's expected embedding space, resolving dimensional mismatches while avoiding explicit reintroduction of attention mechanisms that could mask representational deficiencies.

FR-6: End-to-End Pipeline Integrity: The system is required to facilitate continuous forward and backward propagation across tokenization, encoding, adaptation, decoding, and evaluation stages, allowing stable training and controlled experimentation.

3.1.3 Non-Functional Requirements

The non-functional requirements establish evaluation criteria in line with the diagnostic instead of optimization-oriented goals of the thesis.

NFR-1: Factual Sensitivity: The evaluation framework should be receptive to factual inconsistencies, hallucinations, and omissions, even when surface-level fluency and ROUGE scores are high. The choice of metrics should therefore comprise of fact-conscious measures, besides the lexical overlap.

NFR-2: Controlled Computational Setting: To make sure that the representational properties contribute to the observed differences than resource disparities, all experiments have to be performed with the identical hardware, batch size, and optimization settings.

NFR-3: Long-Context Stress Testing: The architecture should be tested at increasing lengths of the input to examine the hypothesis of whether SSM compression happens to exacerbate factual degradation as contextual load increases.

NFR-4: Modularity for Ablation: The system design should facilitate ablation of the adapter layer and substitution of alternative projection mechanisms to allow isolation of whether failures originate in the SSM encoder or in encoder-decoder interface.

3.1.4 Constraints and Assumptions

Hardware Constraints: Training and evaluation are performed in a single-GPU setup, encouraging careful control of parameter counts and batch sizes but not serving as the primary measure of success.

Tokenization Constraint: A shared BERT-compatible WordPiece tokenizer is used across encoder and decoder to prevent vocabulary-induced confounds in reconstruction fidelity.

Dataset Constraint: XSum is used as the primary evaluation corpus due to its document-level abstraction and factual density, which highly-appropriate to detect the failures in reconstruction.

Modeling Assumption: The decoder is assumed to be sufficiently expressive such that the observed factual degradation can be attributed mainly to limitations in the encoder's latent representations rather than the inability of the decoder.

3.1.5 Compliance with Standards and Best Practices

The project adheres to established standards for reproducible and responsible machine learning research, including transparent reporting of experimental settings, modular system design, and compliance with dataset governance principles. Ethical AI guidelines are followed to guide the evaluation of factual incorrectness and misrepresentation risks, especially in summarization tasks.

3.2 Societal Impact

Although automatic summarization has substantial societal benefits, this research identifies a critical risk and that is fluent but factually unreliable summaries generated by compression-heavy architectures.

3.2.1 Positive Implications of Diagnostic Research

By identifying failure modes in hybrid architectures, this work contributes to safer deployment of summarization systems in education, journalism, governance, and science. Understanding where and why factual degradation occurs enables practitioners to design safeguards rather than relying on misleading surface-level metrics.

3.2.2 Risks of Factual Degradation

In high-stakes areas, summaries that seems coherent yet misrepresent facts can cause higher level of misinformation, distort public understanding, and diminish trust in automated systems. The findings of this thesis underscore the inadequacy of fluency-based evaluation and encourage more factual validation pipelines.

3.2.3 Ethical Framing

By undertaking this study, summarization systems are presented as supporting tools requiring human oversight, particularly when it is applied to aggressive compression regimes. The work is in line with the new regulatory initiatives that focus on transparency, auditability, and accountability of AI-generated content.

3.3 Environmental Impact

The motivation for exploring SSM-based architectures is directly related to sustainability concerns, since quadratic attention of Transformers cost causes significant environmental overhead.

3.3.1 Efficiency-Fidelity Trade-off

SSMs offer near-linear computational complexity and reduced memory usage. However, this thesis demonstrates that such efficiency may come at the cost of representational fidelity. The environmental advantage of quick training should thus be measured against the downstream cost of unreliable outputs, particularly in cases where human verification is required.

3.3.2 Sustainable AI as a Holistic Objective

This work argues that sustainability should not be defined by FLOPs or energy consumption alone, but must also by information efficiency: the ability to preserve and transmit necessary factual information per unit of computation. Energy-saving models that undermine truthfulness run the risk of shifting costs rather than eliminating them.

3.3.3 Long-Term Perspective

By rigorously examining the limits of SSM-based context compression, this thesis contributes to a more responsible path toward green AI-one that balances efficiency with faithfulness and recognizes that not all sequence modeling gains translate into reliable downstream reasoning.

Chapter 4

Design Specification and Data Corpus

4.1 Data and Preprocessing Pipeline

4.1.1 Data Corpus

The performance and generalization capabilities of large-scale neural models are dependent on the quality, scale, and diversity of the data upon which they are trained [71] [63]. LLMs are only as good as the data corpus they are trained on. Modern LLMs are trained on Web size data corpuses that encompass all human written texts on the internet. These include books, articles, social media posts, newspapers, scientific journals and a lot more. The quality of these texts are directly transferred to our models. That is why is it important that we carefully select the pre-training and fine-tuning datasets that will be used for training our models. This issue is largely studied and there is a wide variety of large scale open source data corpuses that are available for training LLMs [35] [55] [87]. Each differ from each other in scale, qualities, languages and their nuances. This section provides a meticulous examination of the data corpora that underpin this study. We discuss the pretraining foundational datasets that embed our LLM with general linguistic and world knowledge, afterwards we discuss the specialized fine-tuning datasets that help the model adapt to our specific task of text summarization.

Foundational Pretraining Corpora

To ensure a controlled and interpretable comparison between encoder and decoder representations, both components of the hybrid architecture used in this thesis are initialized from models pretrained on the same large-scale corpus: the Colossal Clean Crawled Corpus (C4) [55]. This design choice isolates architectural and representational effects from data distributional differences, allowing observed failures in factual reconstruction to be attributed primarily to differences in sequence modeling and compression mechanisms rather than pretraining data.

The Colossal Clean Crawled Corpus (C4)

C4 [55] is a large-scale English-language corpus derived from Common Crawl, a publicly available web scrape. The primary English split of C4 comprises approxi-

mately 800 GB of text, corresponding to roughly 156 billion tokens, while the full multilingual version spans several terabytes.

The "Colossal Clean" name refers to a heuristic-based filtering framework that was applied to raw Internet Scraped data. This process includes profanity removal, elimination of code-like content (e.g., documents containing curly braces), language identification using the "langdetect" library to retain English-only text and additional filtering based on document length and structural quality indicators. These filtering operations significantly raise the usability index of such a data corpus as compared with raw web data, but are rule-based and and approximate. This helps model performance by a large factor.

Regardless of its scale, C4 is known to have significant limitations. The heuristic cleaning pipeline was demonstrated to strip text that is related to some dialects, such as African-American English (AAE) and also content that is related to specific demographic identities and this results in representational bias in pretrained models. Furthermore, C4 has large overlap with popular NLP benchmark datasets, which poses a threat of contaminating the data and hence on the downstream evaluation results. These issues are particularly relevant for summarization tasks, where models may appear to perform well but in reality they are testing on data that they have seen before.

T5 Pretraining on C4

The Transformer-based decoder used in this study follows the Text-to-Text Transfer Transformer (T5) architecture [71]. T5 combines all natural language processing tasks into a unified text-to-text format and has become a standard baseline for generative tasks such as abstractive summarization.

The T5 model we use for our research was pretrained on C4 using a span-corruption (text infilling) objective [71]. During pretraining, contiguous spans of tokens are removed from the input sequence and replaced with sentinel tokens, and the model is trained to autoregressively reconstruct the missing spans. This objective explicitly encourages the model to learn long-range dependencies and to recover missing information from surrounding context. As a result, T5 develops strong generative and reconstruction capabilities, making it a suitable decoder for evaluating whether compressed encoder representations retain sufficient factual information for faithful generation.

Hydra Pretraining on C4

The encoder component of the hybrid summarization model is based on the Hydra Structured State Space Model (SSM) architecture [83] and is likewise pretrained on C4. Hydra uses the sequential state update mechanism of Mamba, resulting in bidirectional linear-time processing of long input sequences without the use of self-attention.

Even though Hydra is exposed to the same pretraining data distribution as T5, its learning dynamics are different. Rather than reconstructing masked spans, Hydra

learns to iteratively compress incoming tokens into a latent state. This state-based formulation prioritizes efficiency and scalability but does not explicitly optimize for recoverability of specific factual spans. Consequently, even when pretrained on identical data, Hydra and T5 are likely to encode contextual and factual information in fundamentally different representational forms.

This distinction is central to the hypothesis explored in this thesis, that SSM-based encoders, despite seeing the same corpus as Transformer models, may compress contextual information in a manner that is not fully decodable by downstream Transformer-based decoders.

Fine-Tuning Datasets for Summarization

Following pretraining, the model must be fine-tuned on the specific task data for benchmarking. For text summarization, this means training on document-summary pairs. There are many categories of Summarization Datasets.

Corpus	**Domain**	**Scale** (#Pairs)	**Avg. Length**	**Avg. Summary Length**	**Summary Style**
CNN/DailyMail	News	∼312k	781	56	Mixed
XSum	News	∼227k	431	23	Abstractive
XL-Sum	News	∼1.35M	Varies	Varies	Abstractive
Newsroom	News	∼1.3M	658	26	Mixed
Multi-News	News (MDS)	∼56k	Varies	264	Abstractive
DialogueSum	Dialogue	∼13.5k	131	23	Abstractive
SAMSum	Dialogue	∼16k	94	20	Abstractive
MediaSum	Dialogue	∼464k	1,554	14	Abstractive
AMI Corpus	Dialogue	137 meetings	4,757	322	Mixed
BillSum	Legal	∼23k	2,200	200	Extractive
BigPatent	Legal	∼1.3M	3,500	120	Abstractive
Gov-Report	Government	∼19.5k	∼9,000	∼450	Abstractive
arXiv/PubMed	Scientific	>200k	∼5,000	∼220	Abstractive
SciTLDR	Scientific	∼3.2k	∼5,000	22	Abstractive
MS^2	Scientific (MDS)	∼20k	Varies	Varies	Abstractive
Reddit TIFU	Community	∼120k	500	20–50	Abstractive
WikiHow	Community	∼230k	Varies	Varies	Mixed

Table 4.1: Comparison of Major Text Summarization Corpora

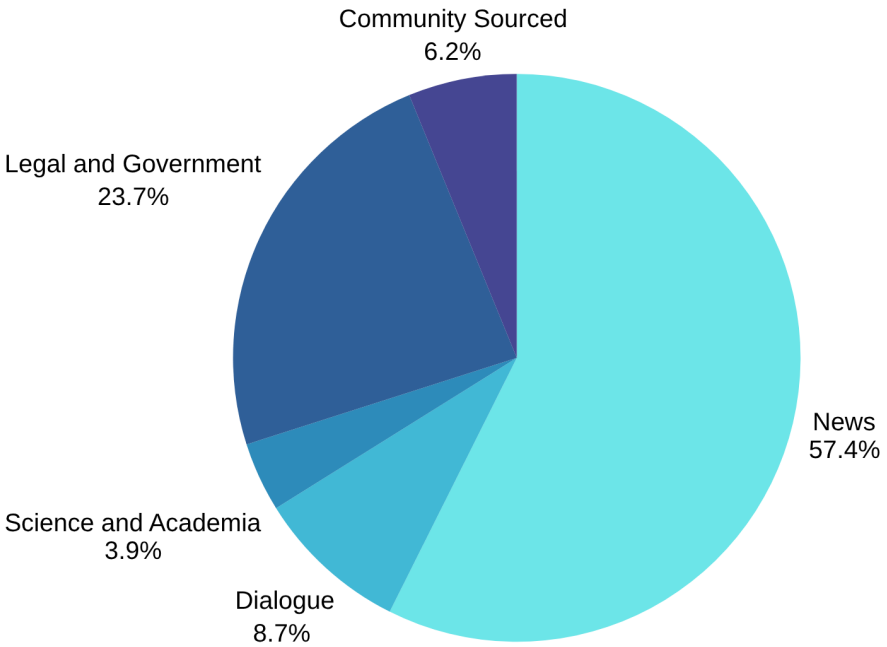

Figure 4.1: Number of Samples in Each Category.

A. General & News Domain

These datasets are the most common benchmarks for single-document summarization and are primarily composed of news articles.

- **CNN/DailyMail:** This dataset [9] is made of news articles paired with multi-sentence summaries. The summaries are somewhat abstractive but often they can include significant extractive sentence fragments. With an average article length of approximately 781 words and an average summary length of 56 words, it is one of the foundational benchmarks for abstractive summarization [9].

- **XSum (Extreme Summarization):** Consisting of 226,000 BBC articles [20], XSum is paired with human written, single-sentence summaries. It is designed for extreme summarization(extreme indicating the difficulty level), a task that demands high compression and a highly abstractive style, as extractive methods perform poorly on this benchmark [20].

- **XL-Sum:** This is a large-scale multilingual dataset [56] made of 1.35 million article-summary pairs from BBC News, covering 44 different languages. The summaries are highly abstractive, making it an important benchmark for evaluating cross-lingual and low-resource summarization capabilities.

- **Newsroom:** This is a large dataset [36] made of 1.3 million articles and summaries from 38 major news publications from 1998 to 2017. Its summaries are known for their stylistic diversity. This dataset combines both abstractive and extractive strategies to reflect real-world editorial practices.

- **Multi-News:** This is the first large-scale dataset for multi-document summarization (MDS) [24], containing news articles from over 1,500 sources and human-written summaries from newser.com. This task requires models to

extract information from multiple, often overlapping or conflicting, sources into a single coherent summary.

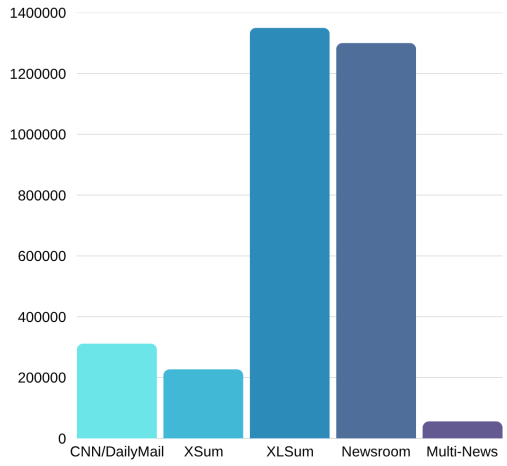

Figure 4.2: Distribution of Samples in News and General Domain.

B. Dialogue & Conversational Domain

Summarizing conversations presents unique challenges, including handling informality, multiple speakers, and disjointed information flow.

- **DialogueSum:** This is a large-scale dataset [54] of 13,460 dialogues from real-life scenarios such as daily life, work, and shopping, accompanied by manually labeled summaries and topics. The summaries are written in formal language from an observer's perspective, requiring the model to abstract away from the conversational style.

- **SAMSum:** This dataset [25] contains over 16,000 messenger-like chat dialogues with manually created, abstractive summaries. The dialogues are typically more informal and shorter than those found in DialogueSum.

- **MediaSum:** As the largest public dialogue summarization dataset [62], MediaSum contains over 463,000 transcripts of media interviews from NPR and CNN. It is characterized by complex, multi-party conversations and is an order of magnitude larger than other dialogue datasets, providing a rich resource for training robust conversational summarizers.

- **AMI/ICSI Meeting Corpus:** These are a suite of multi-modal datasets [4] made of recorded meetings that include audio, video, and orthographic transcriptions. The AMI corpus consists of 100 hours of meetings. It includes both abstractive and extractive summaries. The ICSI corpus contains approximately 70 hours of natural meetings with annotations for dialogue acts and topics. These can be easily leveraged for summarization tasks.

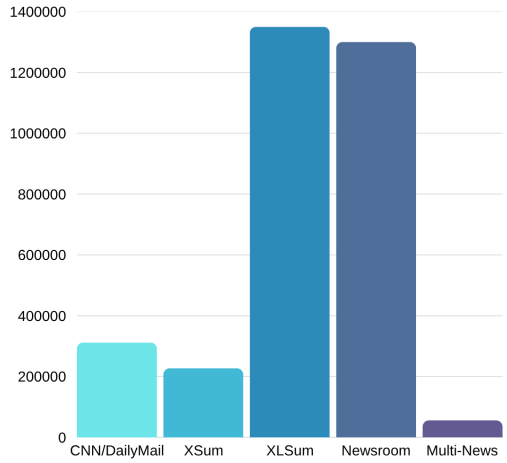

Figure 4.3: Distribution of Samples in Dialogue and Conversation.

C. Legal & Governmental Domain

These domains feature long, highly structured documents with specialized vocabulary, where factual accuracy is paramount.

- **BillSum:** BillSum [22] was the first dataset created for summarizing legislative documents. It contains over 23,000 US Congressional and California state bills with human-written reference summaries.
- **BigPatent:** BigPatent [30] is a large-scale dataset of 1.3 million U.S. patent documents paired with human-written abstractive summaries. It is notable for it's richer discourse structure and more evenly distributed type of content, introducing a different set of challenges for summarization models.
- **Gov-Report:** This is a dataset specifically for long document summarization, consisting of reports from U.S. government research agencies. It features significantly longer documents and summaries than other benchmarks, requiring models to process, understand and remember information from a much more extensive context.

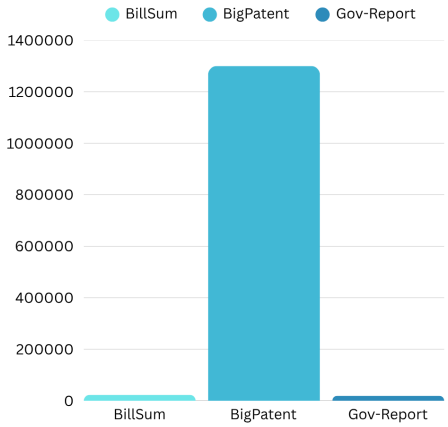

Figure 4.4: Distribution of Samples in Legal and Government.

D. Scientific & Academic Domain

- **arXiv & PubMed:** These two datasets [16] consist of scientific articles paired with their abstracts, warranting deeper understanding for long document summarization in the general scientific and biomedical fields, respectively.
- **S2ORC-Summ:** The Semantic Scholar Open Research Corpus (S2ORC) [40] is a massive corpus of over 81 million academic papers. While primarily a corpus for pretraining, its structure of full papers with abstracts makes it a valuable resource to us for training and evaluating scientific summarization models.
- **SciTLDR:** Similar to XSum, this is a dataset [33] for the extreme summarization of scientific papers, containing over 5,400 "Too Long; Didn't Read" (TLDR) summaries for 3,200 Science papers. It is a multi-target dataset, featuring both author-written TLDRs and summaries derived from peer reviews. This dataset not only demands a high compression rate but also an expert-level understanding of the source material.

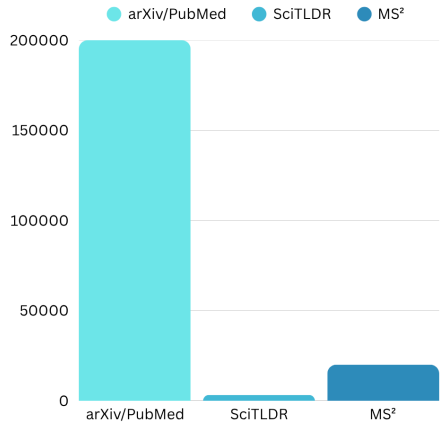

Figure 4.5: Distribution of Samples in Science and Academic Domain.

E. Large-Scale & Community-Sourced Domain

These datasets leverage community-generated content, which often features a more informal and diverse style of writing and summarization.

- **TLDR: Reddit Summaries (e.g., TIFU):** The Reddit TIFU dataset [15] contains over 120,000 posts from the "Today I F'd Up" subreddit, where users write a post and a corresponding TL;DR summary. This dataset is notable for being one of the first to use non-formal, conversational text for abstractive summarization.
- **OpenAI TL;DR:** This dataset [43] was released by OpenAI for training reward models and contains summaries from Reddit's TL;DR posts as well as from CNN and Daily Mail articles, along with human preference data indicating which of several candidate summaries is better.
- **WikiHow:** A large-scale dataset [17] of over 230,000 how-to articles and summary pairs from the WikiHow website, characterized by a diverse range of writing styles from different human authors.
- **Massive Summarization (MS^2):** This is a multi-document summarization dataset [23] in the biomedical domain, consisting of over 470,000 documents and 20,000 summaries derived from scientific literature reviews. It is specifically designed to facilitate the development of systems that can aggregate evidence, including contradictory findings, across multiple studies.

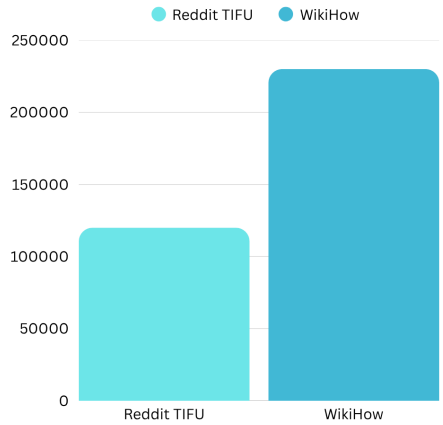

Figure 4.6: Distribution of Samples in Community and Large Scale.

4.1.2 Dataset Creation and Tokenization Pipeline

LLMs are trained on massive text corpora containing terabytes of data from diverse sources such as: web scrapes, books, wikipedia, academic articles, news articles, journals, code repositories etc [35] [55]. Raw text has a large amount of noise, heterogeneity and often duplicate information. It is essential to process and clean these data as the performance of LLMs are largely dependent on the quality of its training data [87].

A. General LLM Preprocessing Pipeline

Before tokenization, raw text undergoes a series of cleaning and normalization steps to standardize the corpus and remove noise that could impede learning.

- **Data Collection and Extraction:** Data collection has been done through scraping internet with bots, collecting through APIs and data archives. PDFs are extracted with relevant tools and HTML links are filtered by stripping away non-textual components and keeping the relevant textual data.
- **Data Cleaning:** This involves removing structural artifacts and irrelevant content. For web-scraped data like C4, the data cleaning process involved below steps:
  - **Boilerplate Removal:** Use heuristics and libraries like jusText, Trafilatura, or rule-based filters.
  - **Language Filtering:** Identify language with tools like FastText or langdetect.
  - **Document Length Filtering:** Remove extremely short or extremely long documents since those have a high probability of being noise.
  - **Profanity / Toxic Content Filtering:** Detect hate speech, profanity, sensitive data, or harmful content using classifiers.
  - **Deduplication:**

1. Exact deduplication: Hash-based (e.g., MinHash).
2. Near-deduplication: SimHash, LSH, or embedding similarity to reduce repeated documents and phrases.

- **Text Normalization:** Normalization ensures textual consistency across the corpus. For our pretraining data corpora this includes Unicode normalization (e.g., NFC) to handle characters with multiple representations, lowercasing (though this can be detrimental if case carries semantic meaning, such as for proper nouns), and handling of extra whitespace and special characters.

- **Deduplication:** As discussed by the creators of the SlimPajama dataset [87], removing duplicate or near-duplicate documents is crucial for training efficiency and preventing the model from overfitting to repeated data. This is often performed at a document or paragraph level using techniques like MinHash [1] to identify similar content.

- **Data Formatting:** For large-scale training, data is typically stored in efficient formats like JSON Lines (JSONL) or Apache Parquet. These are optimized for distributed processing and streaming of big data.

B. Subword Tokenization Algorithms

Tokenization is the process of segmenting text into a finite vocabulary of tokens that an LLM can understand. Modern LLMs universally employ subword tokenization [14] [19].This approach effectively handles rare or out-of-vocabulary (OOV) words and allows for a controlled vocabulary size. Tokenization is one of the most important parts of data preprocessing because the quality of tokenization strongly affects compression efficiency, vocabulary size, and downstream performance. Tokenization is the first thing that has to be accomplished before textual data passes into the input layer of our LLMs. There are multiple algorithms used for tokenization. Our encoder-decoder architecture consists of multiple types of SSM and transformer variant which used different tokenization algorithms during their pretraining process

- **Byte-Pair Encoding (BPE):**
  - **Mechanism:** BPE [14] is a data compression algorithm adapted for tokenization. The training process is iterative and greedy. It begins with a vocabulary of individual characters (or bytes, in byte-level BPE) and repeatedly counts all adjacent token pairs in the training corpus. In each step, it merges the most frequent pair into a new, single token that is then added to the vocabulary. This process continues for a predetermined number of merges, which defines the final vocabulary size.
  - **Use Case:** BPE is famously used by the GPT family of models [63].

- **WordPiece:**
  - **Mechanism:** Used by BERT and its derivatives [37] [41], WordPiece [7] is similar to BPE in that it starts with a base vocabulary of characters and iteratively builds a larger subword vocabulary. The key difference lies in its merge criterion. Instead of merging the most frequent pair,

WordPiece merges the pair that maximizes the likelihood of the training data. This is calculated with the score:

$$\text{Score} = \frac{\text{frequency of pair}}{\text{frequency of first element} \times \text{frequency of second element}}$$

This prioritizes merges where the constituent parts are relatively infrequent on their own. During tokenization of new text, it employs a greedy longest-match-first strategy, finding the longest subword in the vocabulary that matches the beginning of the current word.

- **Use Case:** WordPiece was developed by Google and is notably the tokenization algorithm used to pretrain BERT. It has subsequently been implemented in numerous Transformer-based models derived from BERT, including DistilBERT [41], MobileBERT [44], Funnel Transformers [34], and MPNET [42].

- **Unigram Language Model:**
  - **Mechanism:** In contrast to the additive, bottom-up approach of BPE and WordPiece, Unigram [18] is a subtractive, top-down algorithm. It starts with a very large vocabulary of candidate subwords (e.g., all words plus common substrings) and iteratively removes tokens. In each step, it calculates the loss (e.g., a decrease in the log-likelihood of the corpus) that would be incurred by removing each token and prunes a percentage of the tokens that cause the smallest loss. This process continues until the desired vocabulary size is reached.
  - **Probabilistic Nature:** A key feature of Unigram is its probabilistic foundation. It stores a probability for each token, allowing a single word to have multiple valid tokenization paths. For inference, it typically uses the Viterbi algorithm to find the most likely tokenization sequence for a given word.
  - **Use Case:** Unigram is used by models like T5 and is a core component of the SentencePiece library, which is designed to be language-agnostic and process raw text directly.

Based on these three algorithms many tokenizers have been developed for different LLMs. Some of these popular tokenizers include GPT2 tokenizer, BertTokenizer, llama, mistral tokenizer, falcon, t5-base, gemma-tokenizer, claude-tokenizer. For our decoder-encoder summarizer architecture, we use different SSM-Transformer pair that use different tokenizers.

4.1.3 Benchmarks and Qualitative Evaluation

Due to the qualitative nature of summarization and natural language, no single metric can fully capture the many facets of a good summary. Therefore, we will employ a comprehensive suite of metrics that will be selected by the dimension of quality they assess with the goal of selecting a set of metrics that wholly evaluate the summarizations. This multi-dimensional framework is necessary because of the recognized limitations of individual metrics for summarization. For example, early

reliance on lexical overlap metrics like ROUGE [3] was found to be insufficient, as a summary could achieve a high ROUGE score while being semantically nonsensical or factually incorrect. This led to the development of semantic metrics like BERTScore [51]. Subsequently, it was observed that even semantically plausible summaries could hallucinate facts not present in the source document, a critical failure for summarization. This inspired the creation of dedicated factuality metrics like QAGS [45] and FactCC [26]. A key argument of our thesis is that evaluating a modern abstractive summarizer requires a dashboard of metrics, not a single score. The success of the proposed hybrid model cannot be claimed based on a high ROUGE score alone; it must also demonstrate strong semantic alignment and, most importantly, high factual consistency.

Metric	Assessed Quality	Core Principle	Strengths	Weaknesses
ROUGE-N	Lexical Overlap	N-gram recall between candidate and reference.	Fast, simple, good for content selection.	Ignores semantics, word order.
ROUGE-L	Lexical Overlap	Longest Common Subsequence (LCS) F-score.	Captures sentence-level word order.	Still lexical, less strict than n-grams.
BLEU	Lexical Overlap	N-gram precision with brevity penalty.	Complements ROUGE's recall focus.	Less correlated with human judgment for summarization.
BERTScore	Semantic Similarity	Cosine similarity of contextual token embeddings.	Captures paraphrasing and semantic meaning.	Computationally heavier; bias from base model.
MoverScore	Semantic Similarity	Earth Mover's Distance between embeddings.	Robust to word order; allows many-to-one matching.	Computationally intensive.
QAEval	Factual Consistency	Answering reference-based questions with candidate.	Directly measures information overlap.	Performance depends on QA model quality.
FactCC	Factual Consistency	Binary classification of summary sentence consistency.	Fast, model-based fact-checking.	Weakly supervised; may not cover all error types.
QAGS	Factual Consistency	Consistency of answers to summary-based questions.	Reference-free; provides interpretable error signals.	Complex pipeline; dependent on QG/QA models.
Perplexity	Fluency	Exponentiated cross-entropy of generated text.	Fast, intrinsic measure of model confidence.	Does not measure accuracy or factuality.
CIDER	Consensus/Salience	TF-IDF weighted n-gram similarity.	Rewards salient n-grams appearing across references.	Primarily designed for image captioning.

Table 4.2: Evaluation metrics for Summarization

A. Lexical Overlap Metrics (Content Selection & Fluency)

These metrics evaluate summary quality by measuring the overlap of lexical units (n-grams) between the generated summary and one or more human-written reference summaries.

- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation):** As the de facto standard metric in summarization, ROUGE is recall-oriented,

measuring how much of the reference summary is captured in the generated summary.

  - **ROUGE-N:** Measures the overlap of n-grams (e.g., ROUGE-1 for unigrams, ROUGE-2 for bigrams). Higher-order n-grams are often used as a proxy for fluency.
  - **ROUGE-L:** Measures the Longest Common Subsequence (LCS), which accounts for sentence-level word order without requiring the matching words to be consecutive.
  - **Strengths:** ROUGE is simple, fast to compute, language-independent, and has been shown to correlate reasonably well with human judgments of content selection.
  - **Weaknesses:** Its primary limitation is that it is purely lexical; it cannot recognize semantic similarity (e.g., synonyms, paraphrasing) and can be misleading if a summary is factually incorrect but uses overlapping words.
  - **BLEU (BiLingual Evaluation Understudy):** Primarily used in machine translation, BLEU [2] is a precision-oriented metric that measures n-gram overlap and includes a brevity penalty to penalize summaries that are too short. While less common for summarization than ROUGE, it provides a complementary, precision-focused perspective.

B. Semantic Similarity Metrics (Meaning & Coherence)

These metrics go beyond simple lexical overlap counting by using contextual word embeddings. These metrics compare the semantic meaning of the generated and reference summaries.

- **BERTScore:** This [51] metric computes the cosine similarity between the contextual embeddings (from BERT or a similar model) of tokens in the model generated and reference summary pair. It calculates precision, recall, and F1 scores based on these token-level similarities, This allows it to effectively match paraphrases and semantically similar phrases that other metrics will not capture.

- **MoverScore:** Like BERTScore this metric [31] also uses contextual embeddings but it measures the distance between the two texts using the Earth Mover's Distance (EMD). It allows for many-to-one matching of tokens and presents it as comparison of the minimum cost to transform the generated summary's semantic distribution into that of the reference, thereby capturing semantic similarity at a document level.

C. Factual Consistency & Faithfulness Metrics (Accuracy)

A critical dimension for summarization is whether the generated summary is factually consistent with the source document. These metrics are designed specifically to detect such hallucinations.

- **QAEval (Question-Answering Evaluation):** This [59] reference-based metric represents the information in the reference summary as a set of question-answer pairs. It then attempts to answer these questions using the candidate

summary. The final score is the proportion of questions that can be answered correctly, directly measuring the overlap of factual information.

- **FactCC (Factual Consistency Corpus):** This [39] is a model-based approach that performs a binary classification (CORRECT or INCORRECT) on a summary sentence, conditioned on the source document. It is trained in a weakly-supervised manner on data generated by applying rule-based transformations (e.g., entity swapping, negation) to sentences from source documents to create examples of factual inconsistencies.

- **QAGS (Question Answering and Generation for Summarization):** QAGS [47] is a reference-free metric that first generates questions based on the candidate summary. It then answers these questions using both the candidate summary and the original source document. The similarity of the answers from these two contexts determines the factual consistency score, providing an interpretable signal of where inconsistencies lie.

- **TruthfulQA:** While primarily a benchmark [66] for question-answering models, its methodology of using adversarial questions designed to trigger common misconceptions can be adapted to evaluate the truthfulness of summaries, especially in domains prone to misinformation where a model might replicate falsehoods present in the training data.

D. Other Metrics

- **Perplexity (PPL):** An intrinsic metric that measures how well a language model predicts a sample of text. It is calculated as the exponentiated average negative log-likelihood of a sequence. A lower perplexity indicates that the model is more confident in the generated text's structure and word choices. While not a direct measure of summary quality or factuality, it is a useful indicator of fluency and model certainty.

- **CIDER (Consensus-based Image Description Evaluation):** Though originally developed for image captioning [11], its principle is applicable to text generation. It measures the consensus between a generated sentence and a set of reference sentences by performing a TF-IDF weighting on n-grams. This gives more weight to n-grams that are salient and common across the reference set, rewarding summaries that capture the core consensus information.

4.2 Design and Model Specification

4.2.1 Design Review

The models central to our research are Transformer based models [74] and newly introduced SSMs [81] [83]. These two different principle paradigms in modern sequence modeling architectures offer many different benefits over each other. Our motivation for this study is to extract the benefits from both of these architectures through our SSM-Transformer encoder-decoder hybrid architecture. This section will discuss each of these models' designs, the proposed hybrid model's designs and discuss the inherent technical challenges of such cross-architectural fusion.

SSM-based Models

1. **Mamba & Mamba-2:** Mamba [81] and its successor, Mamba-2 [77], are selective Structured State Space Models that represent a significant breakthrough in sub-quadratic sequence modeling. Their core innovation is the selective scan mechanism (S6) [81], which makes the SSM parameters input-dependent. This allows the model to dynamically modulate its internal state based on the content of each token, giving it the ability to selectively focus on relevant information and forget irrelevant history in a manner that mimics the context-awareness of attention. Mamba-2 further refines this approach based on the theory of State Space Duality (SSD) [77], resulting in an even faster and more efficient algorithm. Such models use standard BPE-based tokenizers, including **eleutherai/gpt-neox-20b**. In this research, Mamba and Mamba-2 establishes the performance baseline for modern SSMs. They demonstrate that these architectures are just as able to get quality that is comparable to Transformers on most language benchmarks.

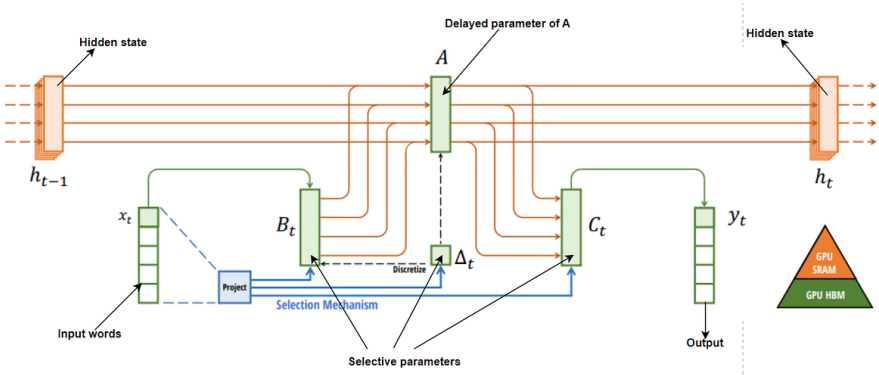

Figure 4.7: Mamba Diagram [81].

2. **Hydra:** Hydra is a variant of the Mamba architecture specifically designed for bidirectional sequence modeling. Whereas standard Mamba is causal and processes sequences unidirectionally, but Hydra processes context simultaneously from both past and future tokens. It achieves this by its core mechanism, a quasiseparable matrix mixer. It offers a mathematically based way of modeling both forward and backward dependencies in a sequence. For our project, Hydra uses a bert-base-uncased tokenizer, which ensures direct compatibility with our proposed Transformer decoder. Hydra is the specific architecture which is chosen for the encoder in our proposed model due to its efficiency and strong performance on language tasks.

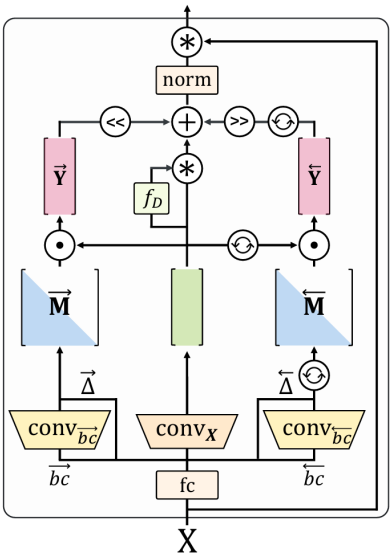

Figure 4.8: Hydra Diagram [83].

Transformer-based Models

1. **T5:** The T5(Text-to-Text Transfer Transformer) [71] model is a versatile Transformer built on the standard encoder-decoder architecture. Its key innovation is the text-to-text framework, which unifies all NLP tasks by reframing them into a single format: a textual input is mapped to a textual output. This is achieved by prepending a task-specific prefix to the input, such as summarize: for summarization tasks, which instructs the model on the desired operation. T5 uses a SentencePiece tokenizer with a Unigram language model basis, which is highly effective for handling a diverse and large vocabulary. T5 serves as a primary state-of-the-art benchmark for our summarization task, representing a powerful, pure-Transformer approach against which we will evaluate our hybrid model's performance.

2. **LLaMA / Llama-3:** The LLaMA family [73] of models are high-performance, open-source, decoder-only Transformer models that represent the cutting edge of large language models. Their architecture incorporates several influential refinements over the original Transformer, including RMSNorm for pre-normalization, the SwiGLU activation function, and Rotary Positional Embeddings (RoPE) for more effective position encoding. These models utilize a highly efficient Byte-Pair Encoding (BPE) tokenizer based on tiktoken. The use of a BPE tokenizer allows for a good balance between vocabulary size and sequence length. In this research, these models serve as a benchmark for overall language modeling capability and demonstrate the performance standard that efficient alternatives must strive to meet.

3. **BERT2BERT:** The BERT2BERT [53] model is a sequence-to-sequence architecture constructed by initializing both its encoder and decoder with weights from a pre-trained BERT model. The decoder is adapted for autoregressive generation by adding cross-attention layers to allow it to attend to the encoder's output representations. This approach leverages the rich linguistic knowledge embedded within BERT for sequence-to-sequence tasks. BERT2BERT uses

the standard WordPiece tokenizer [7] from BERT (bert-base-uncased), which is optimized for English text. The BERT2BERT model is highly relevant as its decoder provides a direct architectural and tokenizer-compatible blueprint for the decoder in our proposed hybrid system.

4. **DistilBART:** DistilBART [41] is a compressed version of the BART model. Its core innovation comes from its creation process: knowledge distillation [10]. In this technique, a smaller student model (DistilBART) is trained to mimic the outputs and internal representations of a larger teacher model (the original BART), resulting in a model that is significantly smaller and faster while retaining a large percentage of the original's performance. DistilBART inherits its BPE-based tokenizer directly from the BART model. It is included as it provides a proof-of-concept for creating efficient sequence-to-sequence models and serves as a benchmark for model compression and efficiency gains.

5. **LOCOST:** LOCOST [76] is representative of a class of efficient Transformer alternatives that aim to overcome the quadratic complexity of self-attention [65]. Its core mechanism involves approximating the full, dense attention matrix using low-rank and sparse matrices. Instead of computing all pairwise token interactions, low-rank methods project the Key and Value matrices to a lower dimension, while sparse methods restrict each token to attend to only a small subset of other tokens. Both approaches reduce the computational complexity from quadratic to linear or near-linear time. The model uses a BPE-based tokenizer from RoBERTa [29]. LOCOST is referenced as an example of an alternative strategy to achieve sub-quadratic complexity within the Transformer paradigm, providing important context for the broader research effort to improve model efficiency.

Model	**Architecture Type**	**Key Innovation/Purpose**	**Role in Research**
Mamba	Selective SSM	Input-dependent parameters for selective state updates.	Foundational selective SSM.
Mamba-2	Selective SSM (SSD)	Faster algorithm based on State Space Duality theory.	State-of-the-art SSM reference.
Hydra	Bidirectional SSM	Bidirectional processing via quasiseparable matrix mixer.	Proposed Encoder Component.
T5	Transformer (Encoder-Decoder)	Unified text-to-text framework for all NLP tasks.	Primary Benchmark Model.
Llama-3	Transformer (Decoder-Only)	SOTA open-source model with architectural refinements.	SOTA Benchmark Model.
BERT2BERT	Transformer (Encoder-Decoder)	Initialized from pre-trained BERT for seq2seq tasks.	Relevant Architectural Benchmark.
DistilBART	Transformer (Encoder-Decoder)	Distilled version of BART for efficient summarization.	Efficient Model Benchmark.
LOCOST	Transformer (Sparse/Low-Rank)	Approximates attention for sub-quadratic complexity.	Related Work / Alternative.

Table 4.3: Summary of Models and Architectures Discussed

4.2.2 Hydra-T5 Hybrid Architecture

Our research adopts an empirical, architecture-focused methodology to investigate whether State Space Model (SSM)-based encoders can produce latent representations that are faithfully decodable by Transformer-based decoders for factual language generation tasks. While SSMs have demonstrated strong efficiency and scalability advantages for long-context modeling, their reliance on implicit state accumulation rather than explicit token-to-token interactions raises concerns regarding the preservation of fine-grained factual details.

The central hypothesis of this thesis is that SSMs compress contextual information in a manner that prioritizes efficiency over recoverability, and that such compression may result in latent representations that are syntactically usable but factually lossy when decoded by Transformer architectures.

In order to test our hypothesis under controlled conditions, we build a hybrid encoder-decoder language model, named Hydra-T5. In which:

- A pretrained SSM-based Hydra encoder is used in place of the standard Transformer encoder.
- A pretrained T5 Transformer decoder is attached to this encoder

By isolating the role of encoder in representation formation and controlling for decoder expressiveness, tokenizer choice, and pretraining corpus, our methodology ensures a focused comparison into whether failures in factual reconstruction are due to inherent limitations of SSM-based context compression.

Architecture Overview:

The Hydra-T5 model adheres to standard encoder-decoder architecture. The encoder is implemented using a Structured State Space Model (Hydra), while the decoder is based on the Text-to-Text Transfer Transformer (T5).

This design is rarely similar to a set of more recent hybrid architecture designs proposed in recent literature that aim to replace Transformer encoders with more efficient sequence modeling alternatives but use Transformer decoders for language generation.

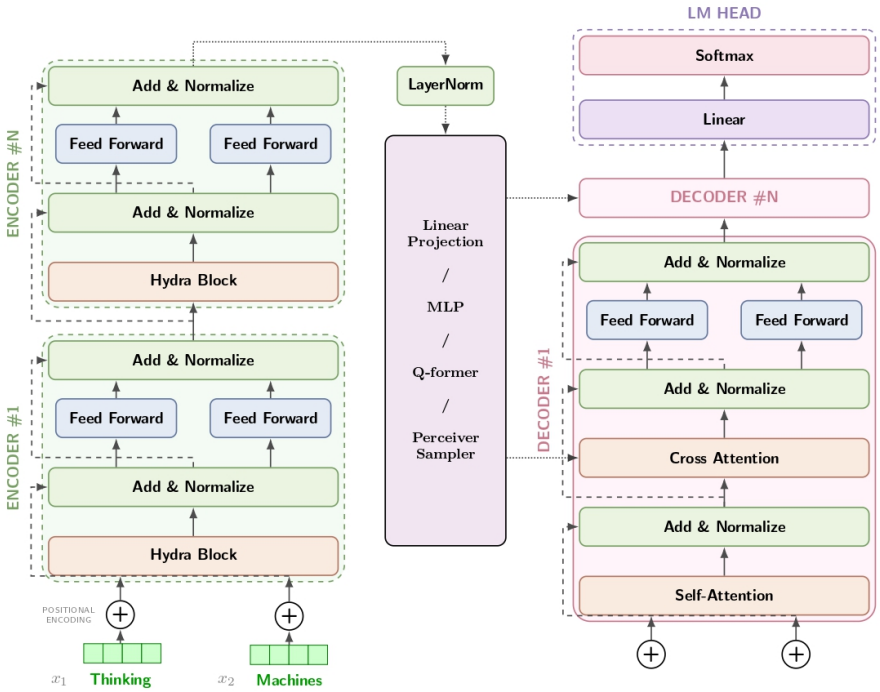

Figure 4.9: Hybrid Hydra-T5 Architecture.

The full processing pipeline is as follows:

1. **Input Processing:** Input tokens are passed through the shared embedding layer to produce dense vector representations for both our encoder and the decoder.

2. **Hydra Encoder:** The dense vectors are processed by a stack of N Hydra blocks. Unlike a Transformer encoder, these blocks do not utilize global self-attention. Instead, they mix information across the sequence dimension using a combination of channel and sequencing mixing state-space transformations.

3. **Adapter Layer:** The final hidden states output by the Hydra Encoder represent the encoded semantic context. These states are projected to match the dimension requirements of the decoder. We experiment with linear projection, Multilayer Perceptron, Q-Former and Perceiver Resampler as our Adapter Layer.

4. **T5 Decoder:** The decoder operates as a standard Transformer. It uses masked causal self-attention to track the generated sequence and, crucially, utilizes a Cross-Attention mechanism to query the Hydra Encoder's output states.

5. **Output Projection and Softmax Layer:** The decoder hidden states are passed through a linear output projection that maps them to the vocabulary dimension. A softmax function is then applied to obtain a probability distribution over the target vocabulary for each time step. During training the model is optimized using a token-level cross-entropy loss between the predicted distributions and the ground-truth target tokens. At inference time the model performs sequence generation by autoregressive standard decoding strategies such as greedy decoding or beam search.

This design enables the decoder to have access to the representations learned by the Hydra blocks. This allows it to be compatibile with pre-training methods like span corruption and masked language modeling.

The Hydra Block (Encoder):

The encoder is constructed with Hydra blocks, a class of SSM-based layers that is optimally designed for efficient long-sequence modeling. Each Hydra block models token sequences as discrete-time dynamical systems, the hidden state of which evolves according to learned transition dynamics.

State Space Formulation

Each Hydra block implements a state space model of the form:

$$s_t = As_{t-1} + Bx_t$$
$$y_t = Cs_t$$

where:

- $s_t \in \mathbb{R}^N$ is the hidden state at time t,
- x_t is the input embedding,
- y_t is the output representation,
- A, B, C are learnable parameters.

In contrast to recurrent neural networks, Hydra is based on parallelizable convolutional implementations of SSMs using diagonal or structured parameterizations of A, enabling efficient computation over long sequences.

Context Compression Behavior

A defining characteristic of Hydra blocks is that context accumulation occurs implicitly through state transitions rather than explicit token-to-token interactions. This results in representations that are:

- Highly compressed
- Order-sensitive
- Dominated by aggregated signals rather than discrete token alignments

Though this compression is beneficial for efficiency, it is a subject of concern when it comes to the preservation of fine-grained factual information, particularly entity relations and precise numeric or referential details.

Stacking and Normalization

Multiple Hydra blocks are stacked to increase model capacity. Each block is followed by:

- Layer normalization
- Residual connections

These design choices mirror those used in Transformer architectures, which ensures that differences in performance can be attributed primarily to representational differences rather than training instability.

The T5 Transformer (Decoder):

The decoder component is based on the Text-to-Text Transfer Transformer (T5) architecture. T5 is a Transformer-based encoder-decoder model originally trained on a unified text-to-text objective.

In this work, only the decoder stack of T5 is utilized. The decoder consists of:

- Masked self-attention layers
- Encoder-decoder cross-attention layers
- Feedforward sublayers

During decoding, the model generates tokens autoregressively:

$$P(y_t \mid y_{<t}, H) = \text{softmax}(W_o z_t)$$

where z_t is the decoder hidden state influenced by cross-attention over the encoder outputs H.

By using a pre-trained T5 decoder, we ensure that:

- The decoding mechanism is expressive and well-calibrated
- Observed factual errors are not due to decoder weakness
- The burden of factual preservation rests primarily on the encoder representations

4.2.3 Architectural Compatibility Challenges

Integrating a State Space Model (SSM) encoder with a Transformer decoder causes a few architectural friction points which are not present in Transformer Encoder-Decoder models. These incompatibilities are analyzed in this section.

A. Tokenizer Mismatch

Our SSM and T5 are trained on different tokenizers, vocabularies, and subword segmentation strategies. T5 uses a SentencePiece-based tokenizer with a fixed vocabulary optimized during pretraining.
To avoid representational inconsistencies:

- We use two tokenizers, each for the respective model.
- Input sequences are tokenized once and passed unchanged to the encoder.
- This ensures consistent token boundaries and embedding alignment.

This design removes tokenizer-induced artifacts and pinpoints representational mismatches to deeper architectural differences.

B. Representation Space Mismatch

Transformer encoders produce representations that are:

- Token-aligned
- Attention-weighted
- Explicitly relational

In contrast, SSM encoders generate representations that are:

- Implicitly aggregated
- Dynamically evolved
- Not explicitly aligned to discrete tokens

As a result, the latent space produced by the Hydra encoder does not naturally project on to the assumptions made by Transformer cross-attention, which expects semantically separable token-level embeddings.

C. Encoder-Decoder Dimensional Discrepancy

One of the main challenges in fusing two different architectures is that the output dimensionality of the encoder does not necessarily match the input dimensionality of the decoder. While in our case both the Hydra encoder and the T5-base decoder share the same hidden size (768), Transformer-based models in general may use different hidden state dimensions. To address this potential mismatch, we introduce an adapter layer that projects the encoder outputs into a representation compatible with the decoder's expected input dimensionality.

4.2.4 Bridge Adapter Mechanism

Adapter Justification

To resolve the representation and dimensional mismatches described in Section 3.3, a dedicated "Bridge Adapter" was introduced between the encoder and decoder. The bridge adapter is a necessary interface between the Hydra encoder and the T5 decoder.

The adapter addresses three critical issues:

- **Project Dimensions:** Map the encoder hidden dimension d_{enc} to the decoder hidden dimension d_{dec}.
- **Harmonize Latent Spaces:** Act as a learnable transition layer that allows the gradients to adjust the SSM representations into a format more digestible for the Transformer decoder's cross-attention heads.
- **Partial re-tokenization of compressed information:** In order for the cross attention in T5 decoder to function, we need to partially retokenize the compressed SSM latent state.

Without the adapter, the decoder receives representations that are syntactically compatible but semantically opaque, resulting in fluent yet factually incorrect outputs.

Bridge Adapter Mechanism Implementation

To investigate whether the observed factual degradation arises from a fundamental incompatibility between SSM-based representations and Transformer decoding, rather than from a specific adapter design, we implement and evaluate multiple bridge adapter mechanisms. Each adapter maps the encoder output representations produced by the Hydra SSM encoder into a latent space that is compatible with the T5 decoder's cross-attention interface. All adapter variants operate on the encoder output sequence $\{h_t\}_{t=1}^{T}$, where $h_t \in \mathbb{R}^{d_e}$, and produce adapted representations $\{\tilde{h}_t\}_{t=1}^{T'}$ suitable for decoding.

Linear Projection Adapter

The simplest adapter is a linear projection that aligns the dimensionality of the encoder output with the decoder hidden size. This adapter is defined as:

$$\tilde{h}_t = W h_t + b$$

where $W \in \mathbb{R}^{d_d \times d_e}$ and $b \in \mathbb{R}^{d_d}$ are learnable parameters, and d_e and d_d denote the encoder and decoder hidden dimensions, respectively. This adapter performs a purely affine transformation and serves as a minimal baseline, allowing us to assess whether dimensional alignment alone is sufficient for faithful factual reconstruction.

MLP-Based Adapter

To introduce non-linear feature remapping, we implement a multi-layer perceptron (MLP) adapter with a single hidden layer:

$$\tilde{h}_t = W_2\, \sigma\left(W_1 h_t + b_1\right) + b_2$$

where $W_1 \in \mathbb{R}^{d_m \times d_e}$, $W_2 \in \mathbb{R}^{d_d \times d_m}$, and $\sigma(\cdot)$ denotes a non-linear activation function. This adapter allows for limited re-structuring of the compressed SSM representations while remaining computationally lightweight. Layer normalization is applied before

and after the MLP to stabilize training and reduce scale mismatch between the encoder and decoder.

Q-Former Adapter

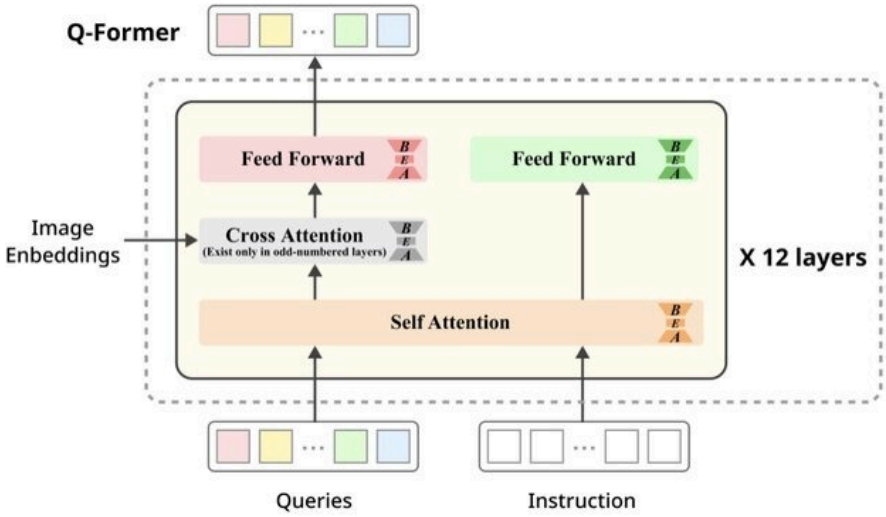

Figure 4.10: Q-Former Diagram [68].

To explicitly model interactions between encoder representations and a fixed set of learnable queries, we adopt a Q-former-based [68] adapter. In this formulation, a small set of trainable query vectors $Q \in \mathbb{R}^{N_q \times d_d}$ attends over the encoder outputs using cross-attention:

$$\tilde{H} = \text{CrossAttn}(Q, H, H)$$

where $H = \{h_t\}_{t=1}^{T}$. The resulting adapted representations $\tilde{H} \in \mathbb{R}^{N_q \times d_d}$ act as a bottlenecked summary of the encoder sequence. This adapter introduces explicit attention-based retrieval of information from the SSM encoder and tests whether selective querying can recover factual content from compressed representations.

Perceiver Resampler Adapter

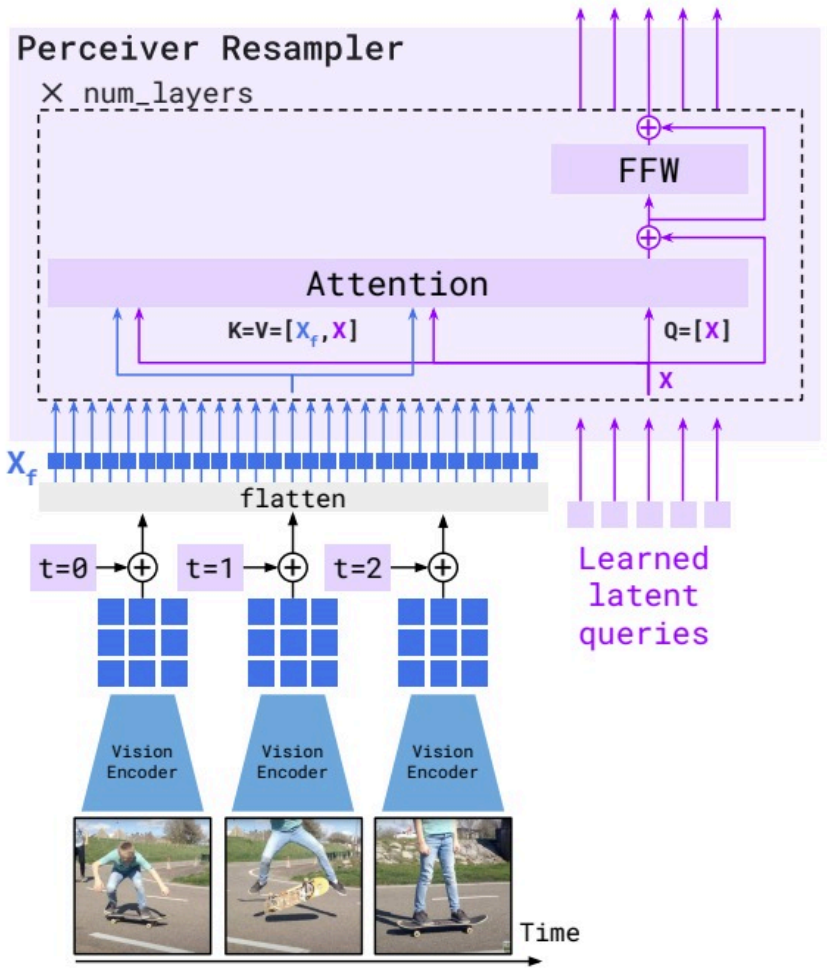

Figure 4.11: Perceiver Resampler Diagram [58].

Finally, we implement a Perceiver-style resampler [58] adapter, which uses a fixed number of latent vectors to repeatedly attend over the encoder outputs. Let $Z \in \mathbb{R}^{N_z \times d_d}$ denote a set of learnable latent embeddings. The resampling operation is defined as:

$$Z' = \text{CrossAttn}(Z, H, H)$$

optionally followed by multiple attention-feedforward blocks. The resulting latent representations Z' are passed to the T5 decoder as encoder states. This adapter enables aggressive sequence length reduction while preserving global information, and allows us to evaluate whether structured resampling mitigates the loss of factual detail introduced by SSM-based compression.

Each adapter tests whether architectural or attention-based retrieval can recover factual content from compressed SSM representations.

4.2.5 Training Objective and Optimization

The model is trained to maximize the log-likelihood of the target text Y given the corrupted input X. The objective function is the standard Cross-Entropy Loss:

$$\mathcal{L}_{MLE} = -\sum_{t=1}^{T} \log P(y_t \mid y_{<t}, X; \theta)$$

Where:

- y_t is the target token at step t.
- $y_{<t}$ represents all previous target tokens.
- X is the hidden representation from the Hydra encoder.
- θ represents the trainable parameters of the hybrid network.

4.2.6 Alignment Challenges in Hybrid SSM-Transformer Training

After integrating the Hydra encoder, bridge adapter, and T5 decoder, the model must learn a shared representational interface that allows meaningful information to flow from the encoder to the decoder. This alignment problem is non-trivial due to the following challenges.

Catastrophic Forgetting and Degenerate Convergence

During early experiments, naïve joint training of all components often resulted in degenerate convergence. In such cases, the model achieved low perplexity and loss values on downstream tasks, yet produced incoherent or semantically meaningless outputs at inference time. This behavior is characteristic of catastrophic forgetting, where gradient updates distort pre-trained representations in a way that satisfies the training objective while destroying previously learned linguistic structure.

This issue became particularly acute when gradients obtained by updating the decoder with randomly initialized adapter layers were used, causing the latent space of the decoder to be quickly corrupted.

Gradient Mismatch Between Architectures

An in-depth analysis of gradient statistics revealed an underlying mismatch between the Hydra encoder and the Transformer decoder. In particular, SSM-based encoders are more likely to generate a gradient of higher magnitude and lower variance across layers, which indicates their dynamical formulation. Transformer decoders, on the other hand, are based on highly normalized gradient flows, which are induced by attention mechanisms, residual connections, and layer normalization.

When these gradients are unrestricted and free to propagate across the adapter boundary, the larger gradients which originate from the SSM encoder can cause

disproportionately large parameter updates in the decoder. This imbalance frequently results in unstable optimization and contributes directly to model collapse.

Randomly Initialized Adapter Layers

Even though the Hydra encoder and T5 decoder are both initialized with pre-trained checkpoints, the bridge adapter layers are instantiated randomly using Xavier initialization [5]. In the early training, this leads to two compounding problems:

- The decoder receives unstructured and semantically meaningless representations from the adapter.
- The encoder receives noisy and uninformative gradients propagated backward through the untrained adapter.

Our empirical testing showed that this bidirectional noise significantly increases the probablity of optimization instability, particularly during the initial stages of training.

4.2.7 Training Stability and Alignment Strategy

To overcome the challenges highlighted above, we develop a training framework that ensures gradual alignment, proper gradient flow, and architectural flexibility.

Progressive Unfreezing Strategy

In order to ensure training consistency and maintain the learned representations of the pretrained backbones, we implement a three-stage Progressive Unfreezing Strategy. Direct end-to-end training of our architecture (SSM-to-Transformer) lead to gradient shock. This is the point at which large error signals corrupts pre-trained weights. We worked around this through the following regime:

- **Stage 1: Adapter Projection Alignment**

  We start our strategy by freezing the parameters of both the SSM encoder and the Transformer decoder ($\theta_{enc}, \theta_{dec}$). We update the parameters of the adapter module (θ_{adapt}) only. This stage bridges the representational gap between the two different architectures. This steps ensures that the gradients flowing through the SSM are compatible with the expected gradients of the T5 decoder.

- **Stage 2: Semantics-to-Signal Alignment**

  Once the adapter is stabilized, we unfreeze the Transformer decoder while keeping the SSM encoder fixed. This stage helps the decoder align its linguistic capabilities with the continuous signal representations provided by the SSM. This does not allow the decoder to ignore the encoder context while adapting to the specific downstream task.

- **Stage 3: End-to-End Optimization**

  In this final stage, all the modules are unfrozen. The entire network is fully updated with parameters. Because the connections between modules have been

pre-aligned in Stages 1 and 2, this stage allows for fine-grained adjustments without the risk of destabilizing the pre-trained features, ensuring optimal convergence.

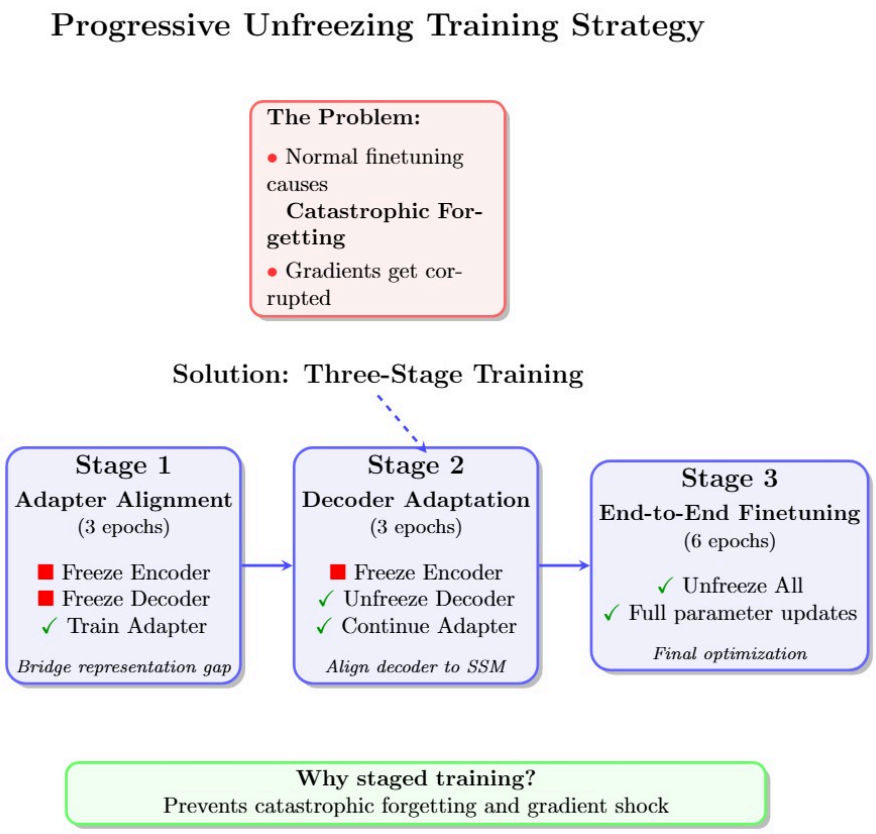

Figure 4.12: Diagram of Three Stage Training.

This staged training procedure ensures that the decoder's linguistic competence is preserved while allowing the encoder-adapter interface to converge to a usable representation.

Flexible Adapter Design

The bridge adapter layers are designed to support both SSM encoder and Transformer decoder configurations. The adapter does not assume matching hidden dimensions between the SSM encoder and Transformer decoder. Rather, all adapter variants directly project encoder outputs into the expected latent Space of the decoder.

This design decision enables the evaluation of alternative encoder widths, sequence lengths and compression strategies with any choice of encoder or decoder. This makes sure that the observed failures cannot be attributed to mismatched dimensionality alone.

Gradient Scaling and Normalization

In order to operate in the presence of gradient imbalance between the SSM encoder and Transformer decoder, we apply layer normalization at the encoder-adapter interface. This kind of normalization ensures that:

- Encoder outputs are rescaled to a distribution that is compatible with Transformer cross-attention.
- Gradients which are propagated into the decoder remain within a stable range.

It was empirically analyzed and thus verified that SSM encoders produce larger and more uniform gradient magnitudes than the Transformer components. Normalizing representations prior to the adapter significantly improves training stability and reduces the probability of optimization collapse.

4.2.8 Summary of Training Framework

Our proposed training framework is designed to enable stable joint of hybrid SSM-Transformer models while retaining the integrity of pre-trained components. By combining selective freezing, adapter-based alignment, and gradient normalization, our framework allows the model to converge consistently without causing catartrophic forgetting.

In the end, even after this carefully controlled training regime, the hybrid model continues to exhibit systematic factual degradation. This observation strengthens the central claim of this thesis that the failure to preserve factual content arises not from training instability or insufficient optimization, but from architectural limitations in the representational properties of SSM-based encoders.

Chapter 5

Results and Analysis

5.1 Baseline

In order to evaluate the performance of the proposed Hydra -T5 hybrid model and empirically test our hypothesis about the failure of the propagation of factual information on the basis of SSM latent representations, it is required that our model is compared with powerful and well-established baselines. Baseline comparisons enable us to contextualize observed performance degradations and ensure that any deficiencies are attributable to architectural differences rather than evaluation artifacts or weak reference models.

Since the decoder part of our proposed model is implemented as the Text-to-Text Transfer Transformer (T5), we take T5 itself as the main baseline. This option gives the opportunity to make a controlled comparison where the decoder architecture, tokenization scheme and training goal are the same, and only the encoder mechanism varies between models. Due to this, any visual differences in the quality of the generated information or factual accuracy can be more reliably explained by the representational quality of the Hydra SSM encoder.

As our baseline model, we use the pretrained T5-base checkpoint. T5-base represents a popular and well-researched configuration that balances capacity and computational efficiency, so it can be compared well to our hybrid architecture. The architectural configuration of the T5-base model is summarized below:

- **Parameter count:** Approximately 220 Million
- **Architecture:** 12-layer Transformer encoder / 12-layer Transformer decoder
- **Hidden size:** 768
- **Number of attention heads:** 12
- **Vocabulary size:** 32,128
- **Pretraining objective:** span-based masked language modeling (text infilling)

During pretraining, T5 employs a span-corruption objective in which contiguous spans of input tokens are replaced with sentinel tokens, and the model is trained to

reconstruct the missing spans autoregressively. This objective encourages the model to capture long-range dependencies and has been shown to be particularly effective for generative tasks such as abstractive summarization. Thereupon, T5 is a robust upper-bound baseline to compare the effectiveness and performance of alternative encoder representations in combination with a Transformer decoder.

We fine-tune the T5 baseline for the same number of training epochs as the proposed Hydra-T5 model on both the XSum and CNN/DailyMail datasets. The same preprocessing steps, tokenization procedures, batch sizes, optimization settings, and decoding configurations are employed across models to ensure a fair and controlled comparison. By aligning the training and evaluation conditions in this way, we minimize confounding factors and gurantee that observed performance differences caused by architectural distinctions, specifically, the replacement of the Transformer encoder with an SSM-based encoder, rather than from discrepancies in experimental setup. Detailed quantitative and qualitative performance comparisons between the baseline and the proposed model are presented in the subsequent sections.

5.2 Attempting at Pretraining a Hydra Infused Vanilla Transformer

In addition to downstream fine-tuning experiments, we conducted an exploratory investigation into the feasibility of pretraining a Transformer model in which the self-attention mechanism of the encoder is replaced by Hydra-based State Space Model (SSM) modules. This experiment was motivated by the hypothesis that some of the representational limitations observed in downstream Hydra-Transformer hybrids might originate during pretraining, rather than arising solely from encoder-decoder integration or task-specific fine-tuning.

Model Architecture:

We consider two model variants for this study:

- **Vanilla Transformer Baseline:** A standard PyTorch implementation of the Transformer architecture proposed in Attention Is All You Need, employing multi-head self-attention in both the encoder and decoder stacks.
- **Hydra-Transformer Hybrid:** A structurally identical Transformer model in which the self-attention sublayers of the encoder are replaced with Hydra matrix-mixing SSM modules. The decoder architecture, including masked self-attention and encoder-decoder cross-attention, is retained without modification. This design isolates the effect of replacing attention-based token interaction with SSM-based sequence mixing during representation learning.

Apart from this substitution, all architectural hyperparameters (hidden size, number of layers, feedforward dimensions, normalization strategy, and residual connections) are kept identical across both models.

Pretraining Data and Setup:

Both models were pretrained from scratch using a mixed-domain corpus totaling approximately 1 billion tokens, composed of:

- **FinePDFs** (∼500M tokens)
- **DCLM Baseline** (∼300M tokens)
- **FineWeb-Edu** (∼200M tokens)

The dataset was split into 90% training and 10% evaluation, and models were trained using a standard autoregressive language modeling objective. Identical optimization settings, batch sizes, and learning rate schedules were employed to ensure comparability.

Evaluation Benchmarks:

In order to evaluate the general language modeling performance, as well as the linguistic competence, we evaluate the pretrained checkpoints with the help of a set of established benchmarks:

- **Perplexity (PPL) on:**
  - Custom 1B-token mixture
  - WikiText-2 [12]
- **Accuracy-based evaluations on:**
  - LAMBADA (long-range dependency modeling)
  - BLiMP (syntactic and grammatical acceptability)
  - SST-2 (zero-shot sentiment classification)
  - COPA (zero-shot commonsense reasoning)

Results:

Model	**PPL** (Custom)	**Acc** (LAMBADA)	**Acc** (BLiMP)	**PPL** (WikiText-2)	**Acc** (SST-2 ZS)	**Acc** (COPA ZS)
Transformer	59.94	0.00097	0.935	124.22	0.5757	0.57
Hydra-Transformer	1.06	0.000	0.6215	1.35	0.5092	0.50

Table 5.1: Evaluation Results on Language Modeling Benchmarks

Although the Hydra-infused model attains very low values of perplexity it equally does not show significant performance on downstream linguistic and reasoning benchmarks. Accuracy on LAMBADA [13] collapses entirely, and performance on BLiMP [48], SST-2 [8], and COPA [6] degrades substantially relative to the vanilla Transformer.

Discussion:

The unusually small perplexity values of the Hydra-Transformer hybrid point towards the degenerate convergence rather than genuine language understanding. Empirical observation showed that the model is able to exploit local statistical regularities without developing semantically grounded, token-aligned representations. Such behavior is consistent with the hypothesis that SSM-based encoders, when trained without attention mechanisms, are likely to over-compress sequence information into latent dynamics that are not well adapted to discrete token-level prediction tasks.

More importantly, this failure mode occurs even in pretraining, even without any form of encoder-decoder communication and downstream fine-tuning. This observation implies that the representational incompatibility is later identified in Hydra-T5 models and is not merely a byproduct of adapter design or training instability, but rather a reflection of a more profound tension between SSM-based sequence mixing and the representational requirements of Transformer-style token decoding.

Implications for the Main Study:

Considering the computational cost of pretraining from scratch and the observed representational collapse, we conclude that full pretraining of Hydra-infused Transformer encoders is neither feasible nor recommended within the constraints of realistic computers. These findings further inspire the methodological approach taken in the rest of this thesis: isolating the representational role of the encoder by pairing a pre-trained Transformer decoder with an SSM-based encoder, and thus allowing targeted evaluation of latent space compatibility without conflating effects from failed pretraining.

5.3 Comparison of Different Adapter Layer Mechanisms

The most important structural component in the proposed Hydra-T5 architecture is the adapter layer. It serves as the sole interface through which representations produced by the SSM-based Hydra encoder are presented to the Transformer-based T5 decoder. Because these two components rely on fundamentally different sequence modeling paradigms, continuous state evolution vs discrete token-level attention, here the adapter layer is responsible for both dimensional compatibility and semantic mediation between heterogeneous latent spaces.

To systematically analyze the extent to which representational incompatibility contributes to factual degradation, we evaluate four adapter mechanisms commonly employed in hybrid or multimodal architectures:

1. **Linear Projection Adapter**
2. **Multi-Layer Perceptron (MLP) Adapter**
3. **Q-Former Adapter**

4. **Perceiver Resampler Adapter**

Each adapter is evaluated under identical training conditions on the XSum dataset, using the same Hydra encoder, T5 decoder, tokenization strategy, and optimization framework. This controlled setup ensures that performance differences can be attributed directly to the adapter mechanism rather than confounding architectural or training variations.

5.3.1 Rouge Metrics for Each Adapter

Linear Projection:

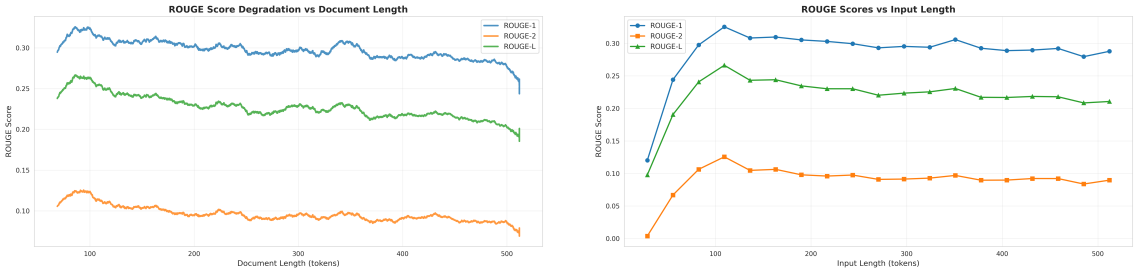

Figure 5.1: Rouge Metrics for Linear Projection

MLP:

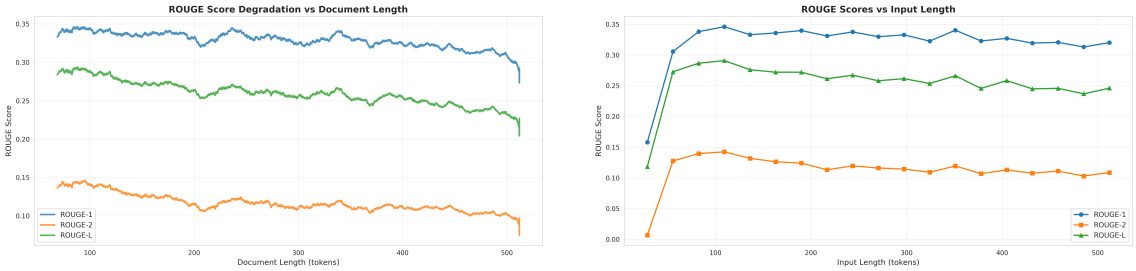

Figure 5.2: Rouge Metrics for MLP Projection

Q-Former:

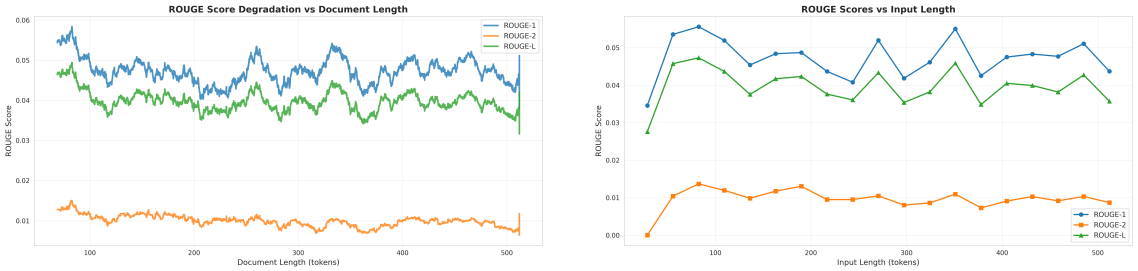

Figure 5.3: Rouge Metrics for Q-Former Projection

Perceiver Resampler:

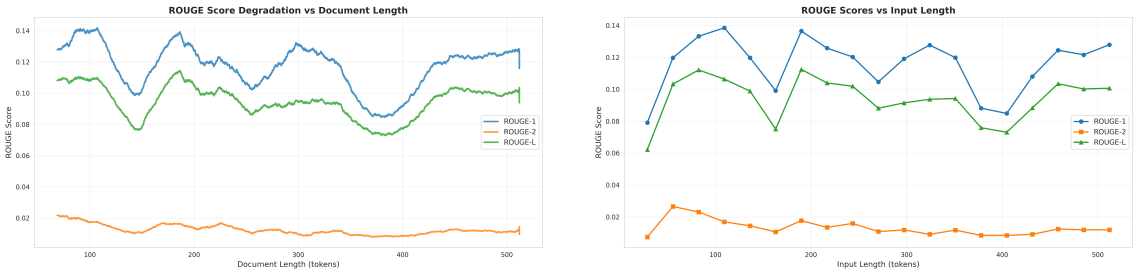

Figure 5.4: Rouge Metrics for Perceiver Resampler Projection

Projection	Projection Parameters	ROUGE-1	ROUGE-2	ROUGE-L	Loss	Avg Length
Linear Projection	589,824	28.27	8.84	21.56	1.88	14.9
MLP	1,181,184	31.38	10.76	24.52	1.95	16.2
Q-Former	2,385,408	4.65	0.97	3.90	1.93	18.6
Perceiver Resampler	9,492,480	11.89	1.26	9.61	5.34	17.8

Table 5.2: Rouge Scores across Different Projection Layers

5.3.2 Summary Quality

Adapter	Reference	Prediction
Linear Projection	England defender Alex Scott has signed a new contract with Arsenal Ladies.	"I have signed" Arsenal Ladies defender Alexa Sanchez.
MLP	Officers who police Scotland's railways are to be armed with Tasers in a bid to increase security on the network.	Officers policing Scotland's railways are preparing to use Tasers to improve security.
Q-Former	A 33-year-old woman has been found guilty of stabbing her boyfriend to death on the balcony of her Rutherglen flat.	((Close) (Glasgow) woman has been convicted of murdering her neighbour.
Perceiver Resampler	A man has been charged with three counts of possessing a bladed article in a public place.	The a a a new man has been found to a new-year-year-year-old man.

Table 5.3: Randomly Picked Summary from each of the Adapter Trained Models

Through both quantitative evaluation and qualitative analysis of generated summaries, we observe a clear separation in behavior between attention-based adapter mechanisms and projection-based adapters. Specifically, the Q-Former and Perceiver Resampler adapters consistently fail to learn a stable and semantically meaningful mapping from the SSM encoder's latent representations to the Transformer decoder's cross-attention space. In practice, these adapters exhibit severe training instability and frequently collapse to degenerate solutions, resulting in either non-convergent training dynamics or fluent but factually ungrounded outputs.

On the contrary, the Linear Projection adapters and MLP-based adapters exhibit a stable training behaviour and are able to mediate the information flow between the Hydra encoder and T5 decoder. Both projection-based adapters prevent catastrophic model collapse and allow the decoder to effectively condition on encoder representations. Among them, the MLP adapter yields the superior performance in all the evalauted metrics considered on the XSum dataset. Specifically, an MLP adapter with three hidden layers achieves the best balance between representational flexibility and optimization stability, providing non-linear properties to restructure compressed SSM representations without loss of training dynamics.

Based on these observations, we adopt the MLP-based adapter as the main bridging mechanism in all the further experiments. This selections is driven by its empirical performance, and also by its ability to introduce controlled non-linearity without making any extra attention-based inductive biases that seem to be incompatible with SSM-generated latent states. The failure of Q-Former and Perceiver-style adapters further supports the broader thesis claim that factual degradation arises from intrinsic representational misalignment between SSM encoders and Transformer decoders, rather than from insufficient adapter expressiveness.

5.4 Training and Validation Loss

5.4.1 Training on XSum Dataset

Here, in this section, we compare the training and validation loss curves of the proposed Hydra-T5 model in comparison with the Transformer-based T5 baseline.

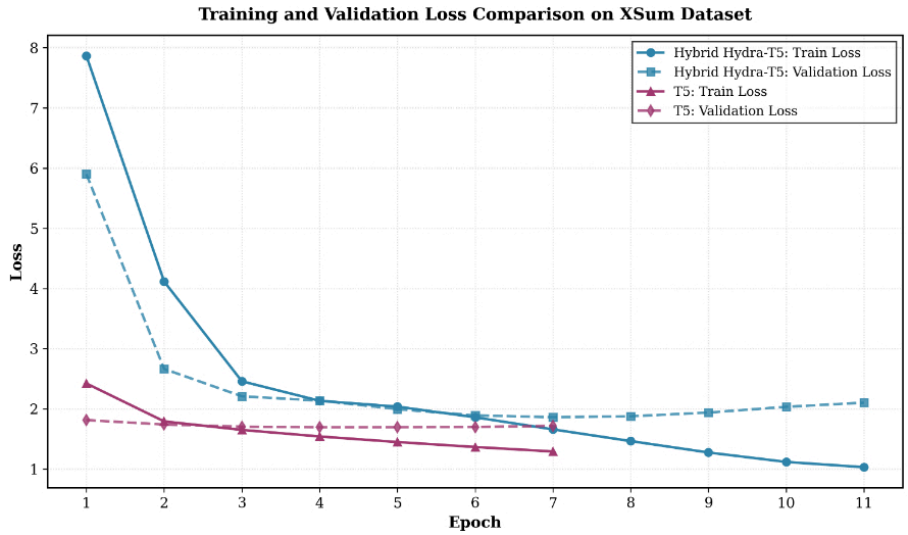

Figure 5.5: Training and Validation Loss for our Model and T5 over XSum Dataset

The Hydra-T5 has a significantly higher initial training (7.87) and validation loss (5.90) compared to T5 (train: 2.42, val: 1.81), and this is due to the challenge of aligning untrained SSM-based encoder representations with a pretrained Transformer decoder. Despite this, Hydra-T5 shows consistent convergence during 11 epochs, and training loss is decreased to 1.03 and validation loss to 2.10.

The convergence pattern reveals rapid loss reduction in early epochs followed by gradual stabilization, which indicates that the gradient can be propagated effectively through the MLP adapter and successful semantic alignment can be done between the encoder and decoder. T5, on the other hand, converges faster and maintains a smaller train-validation gap due to its pretrained encoder-decoder weights, reaching a final training loss of 1.29 and validation loss of 1.72.

To note, Hydra-T5's validation loss plateau above training loss highlights a residual inefficiency in preserving fine-grained factual information, aligning with our hypothesis that SSM-based compression may degrade factual fidelity despite fluent output. The initial large train-validation gap is another indication of the significance of the adapter layer and and progressive unfreezing strategy in stabilizing cross-architecture training.

Overall, these results demonstrate that while Hydra-T5 is trainable and capable of achieving low loss, its performance in factual reconstruction is inherently limited by the representational properties of the SSM encoder, which is why the hybrid design choice and the reliance on a T5 decoder for downstream summarization tasks is justified.

5.4.2 Training on CNN/DailyMail Dataset

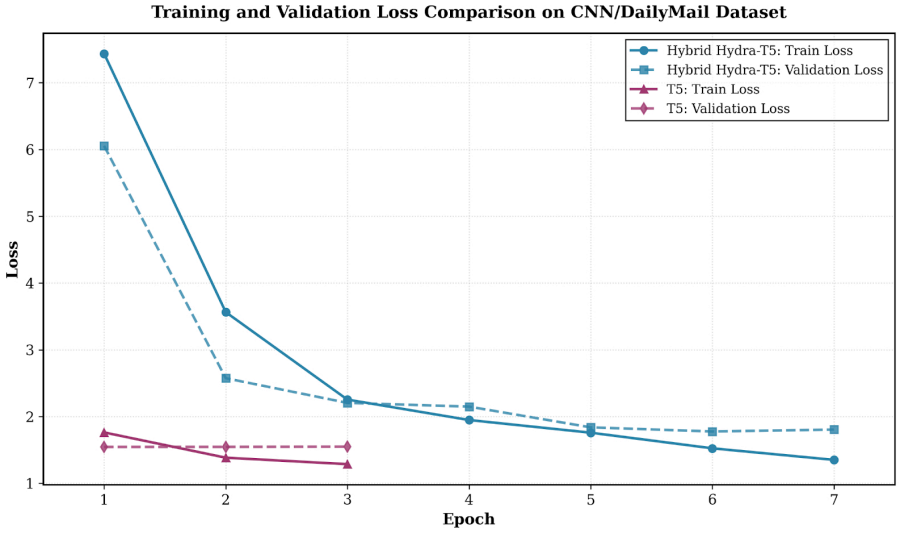

Figure 5.6: Training and Validation Loss for our Model and T5 over CNN/DailyMail Dataset

In comparison to XSum, which requires extreme compression into single sentences, CNN/DailyMail allows for multi-sentence summaries that are less constrained, making it a comparatively easier task. This is seen in the remarkably fast convergence of T5, that within three epochs it has reached a stable performance with minimal training required. The representations provided by the pre-trained T5 model are highly relevant to the CNN/DailyMail summarization task already, allowing it to rapidly adapt without significant architectural modification. In contrast, the Hybrid Hydra-T5 with MLP begins with significantly a larger initial loss (7.44 vs. 1.76), which reflects the need for its hybrid architecture to first align with its representation space before it can effectively converge on the task. Over seven epochs, Hydra-T5 undergoes a steep learning curve and gradually bridges the gap since its multi-head architecture learns to integrate information more appropriately for this summarization format.

5.5 Assessing Language Modeling Capability

To evaluate the quality of text generated by our hybrid Hydra-T5 model, we employ a combination of quantitative and qualitative metrics. During training, we monitor perplexity (PPL) as an indicator of the model's ability to learn fluent and coherent language. After training, we generate summaries for the evaluation datasets and perform manual qualitative assessment on a randomly selected subset to examine coherence, relevance, and factual consistency.

5.5.1 Generated Summaries

Selected examples of generated summaries are presented in the table below. We also include human generated critique of each generated summaries.

Table 5.4: Analysis of Summary Problems

Reference	Prediction	Problem with summary
Officers who police Scotland's railways are to be armed with Tasers in a bid to increase security on the network.	Officers policing Scotland's railways are preparing to use Tasers to improve security.	Too vague, as it says officers are "preparing to use" tasers rather than clearly stating they will be armed with them.
Spain's banks will need an injection of 59.3bn euros to survive a serious downturn, an independent audit has calculated.	Spain is withdrawing large sums of money to support struggling banks as part of a financial aid package.	Incorrect, as the reference states an estimated capital need for banks, not that Spain is withdrawing or providing funds.
Splits within the EU on the relocation of 120,000 migrants have been further exposed as leaders hold an emergency meeting in Brussels.	EU leaders are meeting amid deep divisions over migration policy ahead of a major summit.	Less specific; omits the relocation of 120,000 migrants and misrepresents the emergency meeting as being "ahead of a major summit."
An inquest into the deaths of 11 men at the Shoreham Airshow is not likely for another year, the BBC can reveal.	A second inquest into the Shoreham air disaster has been delayed until next year.	Adds incorrect detail by claiming this is a second inquest and a confirmed delay.
An "extremely rare" signed copy of Adolf Hitler's Mein Kampf is to be sold at auction.	A rare book written by Adolf Hitler is set to be sold at auction.	Too vague; omits that the book is a signed copy of Mein Kampf.
Warner Bros has announced a follow-up to upcoming Harry Potter spin-off film Fantastic Beasts and Where to Find Them.	A follow-up film to Fantastic Beasts and Where to Find Them has been announced.	Omits that the announcement was made by Warner Bros and that it is a Harry Potter spin-off.
Flooding across parts of the UK last winter was the most extreme on record, experts have said.	Flooding caused by winter storms in the UK was the most extreme ever recorded.	Misattributes the cause by stating "winter storms," which was not in the reference.

Continued on next page...

Table 5.4 – continued from previous page

Reference	Prediction	Problem with summary
More than 1,300 militants from so-called Islamic State were killed by British air strikes in Iraq over a 12-month period, according to new figures.	British forces killed hundreds of Islamic State militants during operations in Iraq.	Fails to mention the 12-month period and that the deaths occurred via air strikes.
Security firm G4S has reported a £4m rise in profits to £185m for the first half of 2015 due to new contracts and growth in emerging markets.	Security firm G4S reported increased revenues in the first half of the year.	Misrepresents profits as revenues and omits the profit amount, time frame, and reasons for growth.
A new species of shrimp has been named after Pink Floyd thanks to a pact between prog rock-loving scientists.	A new species has been named in honour of the band Pink Floyd.	Too vague, as it omits that the species is a shrimp and the scientific context behind the naming.
Video replays were used for the first time to send off a player for a professional foul in a Major League Soccer reserve match.	Video replay technology has begun to be used in professional football in the United States.	Overgeneralizes the event by implying widespread professional use rather than a specific MLS reserve match.
British gymnast Brinn Bevan suffered a double leg break while vaulting on Saturday, putting in doubt his hopes of competing at next year's Olympics.	A teenage gymnast suffered a serious double leg break during competition.	Omits the gymnast's identity, nationality, timing, and the impact on Olympic qualification.
Plans to turn a Grade II-listed building in Cardiff Bay into a restaurant and flats have been given the go-ahead by the council.	Plans to convert a Grade II listed building in Cardiff have been approved by councillors.	Omits the specific conversion purpose and the Cardiff Bay location.

Continued on next page...

Table 5.4 – continued from previous page

Reference	Prediction	Problem with summary
Israeli Prime Minister Benjamin Netanyahu has opened a Holocaust exhibition at the Auschwitz Nazi death camp site in southern Poland.	An extremely rare Holocaust-related exhibition has opened at Auschwitz.	Adds incorrect detail about rarity and omits Netanyahu's involvement.
PC gamers will have to wait a little bit longer if they want to play Grand Theft Auto V.	A sequel to Grand Theft Auto V has been delayed for PC gamers.	Adds incorrect detail by claiming a sequel delay instead of a delay to the original game.

5.5.2 Analysis of Generated Summaries

An analysis of the generated summaries shows us some common failures of the model in terms of faithful incorporation and transfer of important information from the source texts. Although the generated outputs are usually fluent and grammatically sound, they tend to lose information, become vague, and distort facts, which compromises on their reliability in summarizing tasks in the real-world.

Loss of specificity is one of the issues noticed. The model has yielded semantically relevant summaries in several occasions that are related to the reference yet they fail to capture important details that constitute the meaning of the original text. For instance, statements such as "officers are preparing to use Tasers" weaken the certainty expressed in the reference, which clearly states that officers will be armed with Tasers. In the same manner, the absence of quantitative information-the transfer of 120,000 migrants or the number of 1,300 militants killed during a period of 12 months creates incomplete summaries. Such omissions make the usefulness of the summaries to be very minimal especially to the news and factual fields where the exactness matters a lot.

Factual distortion or hallucination is another issue which is repeated and in which the model creates something which is not stated in the reference text. Examples include incorrectly framing Spain's banking crisis as an active withdrawal or aid distribution, rather than an estimated capital requirement, and asserting the existence of a "second inquest" when the source only suggests that an inquest is unlikely to occur soon. These inaccuracies suggest that the model occasionally places a high value on producing plausible-sounding stories as opposed to being faithful to the original content, which is a familiar problem in abstractive summarization systems.

Another aspect of the misattribution and causal inferences that has been identified through the analysis is that of misattribution and causal inferences. In one case, the model attributes UK flooding to "winter storms," despite the reference merely stating that experts described the flooding as the most extreme on record. Herein

lies the inclination of this model to come up with causal explanations that are not directly mentioned but instead have to be guessed, and this may easily result in minor yet consistent deviations in the factual truth.

Similarly, contextual dilution can be found in a number of summaries. The contextual qualifiers like the rarity of a signed copy of Mein Kampf or the intervention of the Warner Bros in declaring a film sequel is either diluted or eliminated altogether. Although the summaries are generally accurate at a very low level, the fact that they have lost such contextual attributes makes them less faithful and informative. These vague summaries pose as a big drawback.

5.6 Inference Time

Hydra is a State Space Model (SSM) commonly inspired by the ability to infer faster due to linear-time sequence processing. In order to determine whether this advantage can be converted into practical gains, we compare inference time and throughput between the hybrid Hydra-T5 model and a baseline T5-base model.

Model	**Dataset**	**Avg Inference Time** (sec/sample)	**Throughput** (samples/sec)	**Throughput Gain vs T5** (%)
Hydra-T5 (SSM Enc + MLP)	XSum	0.0860	11.62	+7.8%
T5-base	XSum	0.0927	10.78	–
Hydra-T5 (SSM Enc + MLP)	CNN/ DailyMail	0.2830	3.53	+3.8%
T5-base	CNN/ DailyMail	0.2938	3.40	–

Table 5.5: Inference Performance Comparison

The hybrid Hydra-T5 model consistently achieves slightly lower inference time and higher throughput than the baseline T5 across both datasets. Nevertheless, these performance improvements are very small, which means that SSM-based encoders do not offer significant end-to-end inference acceleration in the case.

This small improvement is attributed to the autoregressive Transformer decoder, which dominates inference cost during summary generation. Since both models share the same T5 decoder and decoding strategy, the efficiency gains from the SSM encoder are limited. Additionally, the MLP projection layer introduces extra computation that partially offsets the encoder-side speedup.

Overall, these results suggest that while SSMs can offer only incremental inference benefits,speed improvements in abstractive summarization require optimization of the decoder or non-autoregressive generation strategies, rather than encoder-only modifications.

5.7 Model Performance

This section evaluates the overall performance of the proposed Hydra-T5 model in comparison to a baseline T5 architecture trained under identical conditions. We have already assessed the generated summaries in section 5.2.2. Now, we evaluate and compare the models on popular metrics for text summarization.

5.7.1 Comparison of Quantitative Metrics

	XSum		**CNN/DailyMail**	
Metrics	**Hybrid Hydra-T5**	**T5**	**Hybrid Hydra-T5**	**T5**
ROUGE-1	0.3138	0.2947	0.2977	0.3051
ROUGE-2	0.1076	0.0987	0.0995	0.1253
ROUGE-L	0.2452	0.2274	0.1935	0.2271
Precision (BERTScore)	0.7091	0.5944	0.5232	0.4546
Recall (BERTScore)	0.6700	0.6440	0.5757	0.4950
F1 (BERTScore)	0.6876	0.6151	0.5471	0.4715
Avg Prediction Length (words)	16.2	26.1	52.1	58.9
Avg Reference Length (words)	21.1	21.1	50.2	50.2
Compression Ratio	0.82	1.32	1.16	1.27
Loss	1.9539	1.7170	1.7756	1.5515
Perplexity	7.06	5.57	5.90	4.72

Table 5.6: Detailed Performance Metrics Comparison

5.7.2 Comparison of Factuality and Hallucination Metrics

This section evaluates the main hypothesis of our thesis: while the hybrid Hydra-T5 model, which consists of an SSM-based encoder and a Transformer decoder, can produce grammatically correct and coherent summaries, it exhibits significant weaknesses in preserving factual correctness.

	XSum		**CNN/DailyMail**	
Metrics	**Hybrid Hydra-T5**	**T5**	**Hybrid Hydra-T5**	**T5**
Hallucination Rate	0.5195	0.5675	0.4222	0.1957
Factual Error Rate	0.5929	0.6677	0.5233	0.2482
Entity Recall	0.2283	0.2445	0.1864	0.2252

Table 5.7: Factual Consistency & Hallucination Metrics

The factuality and hallucination metrics indicate a difference in the performance patterns between the two datasets. On XSum, Hybrid Hydra-T5 achieves superior factual accuracy, with lower hallucination rates (0.5195 vs 0.5675) and factual error rates (0.5929 vs 0.6677) compared to the baseline T5 model. This improvement can be attributed to the single-sentence summary constraint of XSum, where the SSM encoder effectively compresses source information into a compact latent representation that is suitable for generating concise, factually grounded outputs.

This advantage, however, tends to fade off significantly on CNN/DailyMail, which requires multi-sentence summarization. Though the baseline T5 model achieves remarkably low hallucination (0.1957) and factual error rates (0.2482), the Hybrid Hydra-T5 features a substantially higher one (0.4222 and 0.5233, respectively). The difference is further demostrated and evident by entity recall scores in which T5 scores are 0.2252 compared to the entity recall scores of Hybrid Hydra-T5 of 0.1864. This performance degradation in longer summarization tasks supports our original hypothesis: the Hydra SSM architecture struggles to propagate and maintain information fidelity across extended generation sequences, proving that SSM's fail to create latent spaces that can be used to create faithful reconstruction by Transformer decoders.

5.7.3 Effect of Input Length on Factual Error Rate

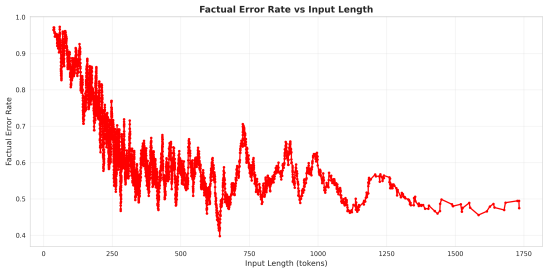

Figure 5.7: For T5 on XSum

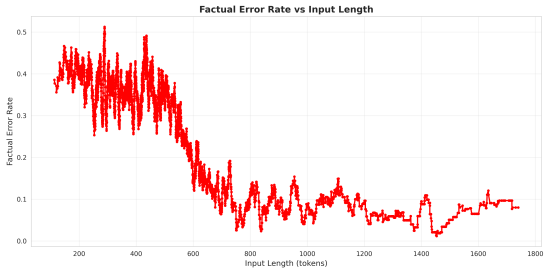

Figure 5.8: For T5 on CNN/DailyMail

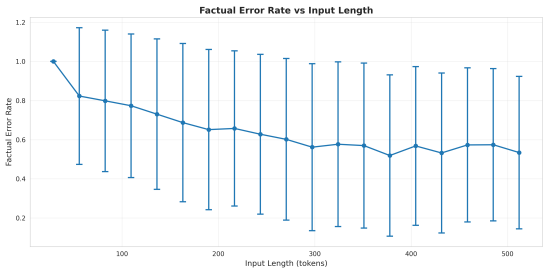

Figure 5.9: For Hybrid Hydra-T5 on XSum

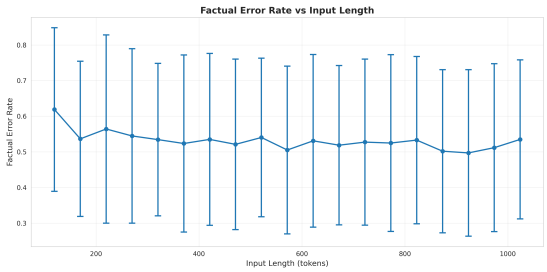

Figure 5.10: For Hybrid Hydra-T5 on CNN/DailyMail

Figure 5.11: Factual Error vs Input Length

We can see that with increased input length, the error rate of T5 decreases drastically. While The error rate for Hydra-T5 remains steady. This shows that while transformer struggles with generating factual and accurate shorter sequence, Hydra-T5 remains incorrectly over any input length.

5.8 Key Findings and Discussion

This part brings together the empirical evidence that Chapter 5 has provided, and gives a critical account of what they imply to our overriding thesis assertion.

5.8.1 Factual Degradation Scales with Reconstruction Complexity

The measures of factuality and hallucination show that the scale of State Space Model (SSM) encoders is dependent on the length of the source documents and the length of the target summaries. Although Hydra-T5 does well on XSum, where high-level abstraction are more valuable, the algorithm performs poorly on CNN/DailyMail, a dataset where a more detailed and fact-preserving summary is more important.

This performance deviation shows the mechanism of compression failure of SSM. The input sequences are compressed by SSM encoders through giving preference to abstract semantic representations and systematic discarding of fine-grained factual information, such as named entities, numerical values, explicit relationships, and discrete events. As such, the resulting latent representations represent what the document is about, as opposed to what the document actually says.

In cases of extreme summarization tasks, where semantic coherence on a high level is needed only, such compression suffices and might be beneficial. But this representational strategy is essentially insufficient in the tasks which require faithful reconstruction of facts. These results provide empirical confirmation of the main hypothesis of this thesis: the propagation of factual context by SSM encoders is not sufficient to be faithfully reconstructed by Transformer decoders.

5.8.2 Adapters Cannot Compensate for Representational Insufficiency

The systematic adaptability analysis in Section 5.3 suggests that factual propagation failures are caused by inadequate information propagation in SSM latent representations rather than by adaptor expressiveness constraints. State-of-the-art attention-based adapters such as Q-Former and Perceiver Resampler were also considered since they had theoretical capabilities of extracting structured information through learned queries and cross-attention. Regardless of this potential, these adapters always did not attain steady training dynamics and often degenerated into degenerate solutions.

The recurrent failure of these attention-based adapters indicates that the information that they are trying to retrieve is not present or is not stored in a form that can be accessed by the attention systems. The fact that this is the case directly disproves the assumption that failures in hybrid architecture is a result of poor adapter design. Along a variety of adapters, between the simple linear projections and complex attention-based modules, factual decay continued as long as the reconstruction complexity was higher than the representational capacity that is conserved by the SSM compression.

Such findings pose that the failure is structural, not architectural: it is not the interface between encoder and decoder that fails, but the fact that the encoder is simply incapable of supporting fact-preserving representations.

5.8.3 SSM Compression Produces Fluency Without Fidelity

Further insight on the ways in which the constraints of SSM encoders are reflected in the output of the generation is made by qualitative analysis of generated summaries (Section 5.6). The summaries that hybrid models generate always have grammatical accuracy, stylistic appropriateness, and semantic relevance to the source documents. Nevertheless, they systematically fail to preserve factual accuracy, displaying recurring patterns of entity substitution, numerical distortion, and omission of critical details.

These errors are not mere random artifacts, rather systematic effects of SSM compression. The continuous state dynamics of SSM encoders tend to favor preservation of smooth, global semantic patterns while eliminating discrete, localized factual. As a result, the latent representations support fluent and coherent text generation but lack the detailed factual grounding required to reconstruct it accurately.

This leads to a characteristic failure mode: the decoder generates text that sounds plausible and contextually appropriate, yet deviates from the factual content of the source. In effect, SSM compression enables fluency but not fidelity.

5.8.4 Limited Efficiency Gains in SSM-Transformer Encoder-Decoder Architectures

Analysis of inference performance (Section 5.7) shows that there are minimal practical benefits to encoder-decoder architectures of the computational efficiency benefits of SSM encoders. Hydra-T5, in comparison to T5 baseline, makes throughput improvements of only 7.8% percent on XSum and 3.8% percent on CNN/DailyMail, substantially lower than what the linear-time complexity of SSMs would theoretically suggest.

This finding challenges one of the reasons behind the adoption of hybrid architectures: better inference efficiency. Even though SSM encoders are more efficient than attention-based encoders at processing long sequences, this is overshadowed in encoder-decoder tasks where autoregressive Transformer decoders dominate overall inference cost.

Therefore, the empirical results indicate that SSM-Transformer hybrid architectures offer limited benefits in terms of efficiency with massive losses in factual consistency, especially when the task requires detailed reconstruction.

Chapter 6

Conclusion

6.1 Limitations and Future Work

6.1.1 Limited Interpretability of SSM Latent Representations

While this work offers strong empirical evidence that SSM-based encoders are unable to propagate fine-grained factual information to a form that can be faithfully decoded by Transformer architectures, the internal structure of SSM latent representations remains largely opaque. We base our conclusions on downstream behavior and evaluation metrics, as opposed to inspection of latent states. Future work could use explainable AI (XAI) techniques, representation probing, or latent state analysis to gain a better insight into what kinds of information are preserved or lost during SSM-based context compression. A deeper understanding of SSM latent space interpretability would be used to other designs of hybrid architectures which more efficiently balance efficiency, fluency, and faithfulness.

6.1.2 Task and Dataset Scope

Our experimental evaluation is primarily conducted on abstractive summarization tasks, specifically using the XSum and CNN/DailyMail datasets. While summarization is a rigorous and appropriate benchmark that can be used to evaluate factual consistency and contextual fidelity, the results might not necessarily be applicable to other downstream tasks, including question answering, dialogue generation or long-form reasoning, where different forms of abstraction may suffice. Future work should extend evaluation to these tasks to determine whether the limitations observed in this study are task-specific or reflection of a broader limitation of SSM-Transformer hybrid architectures.

6.1.3 Computational Constraints and Pretraining Scale

Large-scale hybrid models were out of reach of pretraining from scratch due to computational constraints. As a result, our primary hybrid architecture relies on a pretrained Transformer decoder (T5), which may partially mask or amplify failure modes associated with SSM encoders. Although such a configuration can be seen as realistic low-resource research settings and allows controlled investigation of represen-

tational alignment, it excludes a fully symmetric comparison with Transformer-only baselines pretrained at equivalent scale. Future work using greater computational resources could explore joint or staged pretraining strategies to more thoroughly test the possibility of whether co-adaptation mitigating the representational mismatch as seen in this work.

6.1.4 Adapter Layer Design Space

Although we consider a number of commonly used adapter layer techniques, such as Linear Projection, MLP, Q-Former, and Perceiver Resampler, the larger design space for aligning SSM encoders with Transformer decoders is underexplored. Further specialized or task-aware alignment mechanisms, potentially incorporating auxiliary losses, contrastive objectives, or explicit information preservation constraints, can be used to alleviate the reported degradation in factual faithfulness. Researching such approaches is a good direction to be followed in the future.

6.2 Conclusion

In this thesis, we investigated the practicality of hybrid State Space Model (SSM)-Transformer architectures for abstractive text summarization. We prove, through much extensive empirical analysis, that such hybrids are capable of producing fluent and coherent text, but systematically unable to produce factually faithful summaries, a limitation that we explain by the nature of contextual compression in SSM latent representations. By constructing and analyzing a Hydra-T5 hybrid model that aligns pretrained components with minimal additional training, we isolate representational incompatibilities rather than optimization deficiencies as the primary source of this failure. Given the increasing emphasis on factual reliability and hallucination mitigation in modern language models, alongside the growing popularity of SSMs for efficient long-context modeling, this work provides an important empirical baseline for future research aimed at developing language models that are not only efficient and scalable, but also faithful and trustworthy.

Bibliography

[1] A. Z. Broder, "On the resemblance and containment of documents," in *Proc. Compression and Complexity of Sequences 1997*, 1997, pp. 21–29. DOI: 10.1109/SEQUEN.1997.666900

[2] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02, Philadelphia, Pennsylvania: Association for Computational Linguistics, 2002, pp. 311–318. DOI: 10.3115/1073083.1073135 [Online]. Available: https://doi.org/10.3115/1073083.1073135

[3] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013/

[4] J. Carletta et al., "The ami meeting corpus: A pre-announcement," in *Machine Learning for Multimodal Interaction*, S. Renals and S. Bengio, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 28–39, ISBN: 978-3-540-32550-5.

[5] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, JMLR W&CP, 2010, pp. 249–256.

[6] M. Roemmele, C. Bejan, and M. Gordon, "Choice of plausible alternatives: An evaluation of commonsense causal reasoning," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, European Language Resources Association (ELRA), 2011, pp. 1143–1148.

[7] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 5149–5152. DOI: 10.1109/ICASSP.2012.6289079

[8] R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2013, pp. 1631–1642.

[9] K. M. Hermann et al., *Teaching machines to read and comprehend*, 2015. arXiv: 1506.03340 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1506.03340

[10] G. Hinton, O. Vinyals, and J. Dean, *Distilling the knowledge in a neural network*, 2015. arXiv: 1503.02531 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1503.02531

[11] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2015, pp. 4566–4575.

[12] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," *arXiv preprint arXiv:1609.07843*, 2016.

[13] D. Paperno et al., "The lambada dataset: Word prediction requiring a broad discourse context," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Association for Computational Linguistics, 2016, pp. 1525–1534.

[14] R. Sennrich, B. Haddow, and A. Birch, *Neural machine translation of rare words with subword units*, 2016. arXiv: 1508.07909 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1508.07909

[15] M. Völske, M. Potthast, S. Syed, and B. Stein, "TL;DR: Mining Reddit to learn automatic summarization," in *Proceedings of the Workshop on New Frontiers in Summarization*, L. Wang, J. C. K. Cheung, G. Carenini, and F. Liu, Eds., Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 59–63. DOI: 10.18653/v1/W17-4508 [Online]. Available: https://aclanthology.org/W17-4508/

[16] A. Cohan et al., "A discourse-aware attention model for abstractive summarization of long documents," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, vol. 2, Association for Computational Linguistics, 2018, pp. 615–621.

[17] M. Koupaee and W. Y. Wang, *Wikihow: A large scale text summarization dataset*, 2018. arXiv: 1810.09305 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1810.09305

[18] T. Kudo, *Subword regularization: Improving neural network translation models with multiple subword candidates*, 2018. arXiv: 1804.10959 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1804.10959

[19] T. Kudo and J. Richardson, *Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing*, 2018. arXiv: 1808.06226 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1808.06226

[20] S. Narayan, S. B. Cohen, and M. Lapata, *Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization*, 2018. arXiv: 1808.08745 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1808.08745

[21] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," 2019. arXiv: 1901.02860 [cs.LG].

[22] V. Eidelman, "Billsum: A corpus for automatic summarization of us legislation," in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, Association for Computational Linguistics, 2019, pp. 48–56. DOI: 10.18653/v1/d19-5406 [Online]. Available: http://dx.doi.org/10.18653/v1/D19-5406

[23] A. Fabbri, I. Li, X. She, Y. Sun, and Y. Liu, "Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Association for Computational Linguistics, 2019, pp. 1074–1084.

[24] A. R. Fabbri, I. Li, T. She, S. Li, and D. R. Radev, *Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model*, 2019. arXiv: 1906.01749 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1906.01749

[25] B. Gliwa, I. Mochol, M. Biesek, and A. Wawer, "Samsum corpus: A human-annotated dialogue dataset for abstractive summarization," in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, Association for Computational Linguistics, 2019. DOI: 10.18653/v1/d19-5409 [Online]. Available: http://dx.doi.org/10.18653/v1/D19-5409

[26] W. Kryściński, B. McCann, C. Xiong, and R. Socher, *Evaluating the factual consistency of abstractive text summarization*, 2019. arXiv: 1910.12840 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1910.12840

[27] M. Lewis et al., *Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*, 2019. arXiv: 1910.13461 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1910.13461

[28] Y. Liu and M. Lapata, *Text summarization with pretrained encoders*, 2019. arXiv: 1908.08345 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1908.08345

[29] Y. Liu et al., "Roberta: A robustly optimized bert pretraining approach," in *arXiv preprint arXiv:1907.11692*, 2019.

[30] E. Sharma, C. Li, and L. Wang, *Bigpatent: A large-scale dataset for abstractive and coherent summarization*, 2019. arXiv: 1906.03741 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1906.03741

[31] W. Zhao, M. Peyrard, F. Liu, Y. Gao, C. M. Meyer, and S. Eger, "Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, 2019, pp. 563–578.

[32] I. Beltagy, M. E. Peters, and A. Cohan, *Longformer: The long-document transformer*, 2020. arXiv: 2004.05150 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2004.05150

[33] I. Cachola, K. Lo, A. Cohan, and D. S. Weld, "Tldr: Extreme summarization of scientific documents," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2020, pp. 4766–4777.

[34] Z. Dai, G. Lai, Y. Yang, Q. Le, and R. Salakhutdinov, "Funnel-transformer: Filtering out sequential redundancy for efficient language processing," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

[35] L. Gao et al., *The pile: An 800gb dataset of diverse text for language modeling*, 2020. arXiv: 2101.00027 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2101.00027

[36] M. Grusky, M. Naaman, and Y. Artzi, *Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies*, 2020. arXiv: 1804.11283 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1804.11283

[37] X. Jiao et al., *Tinybert: Distilling bert for natural language understanding*, 2020. arXiv: 1909.10351 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1909.10351

[38] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, *Transformers are rnns: Fast autoregressive transformers with linear attention*, 2020. arXiv: 2006.16236 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2006.16236

[39] W. Kryscinski, B. McCann, C. Xiong, and R. Socher, "Evaluating the factual consistency of abstractive text summarization," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 9332–9346. DOI: 10.18653/v1/2020.emnlp-main.750 [Online]. Available: https://aclanthology.org/2020.emnlp-main.750/

[40] K. Lo, L. L. Wang, M. Neumann, R. Kinney, and D. S. Weld, "S2orc: The semantic scholar open research corpus," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, Association for Computational Linguistics, 2020, pp. 4969–4983.

[41] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. "Distilbart: A distilled version of bart," Accessed: Jan. 1, 2020. [Online]. Available: https://huggingface.co/docs/transformers/modeldoc/distilbart

[42] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mpnet: Masked and permuted pre-training for language understanding," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.

[43] N. Stiennon et al., "Learning to summarize with human feedback," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 3008–3021.

[44] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, "Mobilebert: A compact task-agnostic bert for resource-limited devices," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, Association for Computational Linguistics, 2020, pp. 2158–2170.

[45] A. Wang, K. Cho, and M. Lewis, *Asking and answering questions to evaluate the factual consistency of summaries*, 2020. arXiv: 2004.04228 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2004.04228

[46] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, *Linformer: Self-attention with linear complexity*, 2020. arXiv: 2006.04768 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2006.04768

[47] W. Wang, D. Khashabi, A. Sabharwal, and H. Hajishirzi, "Asking and answering questions to evaluate the factual consistency of summaries," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2020, pp. 5009–5020.

[48] A. Warstadt et al., "Blimp: The benchmark of linguistic minimal pairs for english," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 377–392, 2020. DOI: 10.1162/tacl_a_00321 [Online]. Available: https://doi.org/10.1162/tacl_a_00321

[49] M. Zaheer et al., "Big bird: Transformers for longer sequences," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17 283–17 297, 2020.

[50] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, *Pegasus: Pre-training with extracted gap-sentences for abstractive summarization*, 2020. arXiv: 1912.08777 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1912.08777

[51] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, *Bertscore: Evaluating text generation with bert*, 2020. arXiv: 1904.09675 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1904.09675

[52] T. Zhang, V. Kishore, F. X. Yu, X. Wang, A. J. Smola, and D. Radev, "Factcc: A benchmark for factual consistency in abstractive summarization," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 9332–9346.

[53] C. Chen et al., *Bert2bert: Towards reusable pretrained language models*, 2021. arXiv: 2110.07143 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2110.07143

[54] Y. Chen, Y. Liu, L. Chen, and Y. Zhang, *Dialogsum: A real-life scenario dialogue summarization dataset*, 2021. arXiv: 2105.06762 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2105.06762

[55] J. Dodge et al., *Documenting large webtext corpora: A case study on the colossal clean crawled corpus*, 2021. arXiv: 2104.08758 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2104.08758

[56] T. Hasan et al., *Xl-sum: Large-scale multilingual abstractive summarization for 44 languages*, 2021. arXiv: 2106.13822 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2106.13822

[57] L. Huang, S. Cao, N. Parulian, H. Ji, and L. Wang, *Efficient attentions for long document summarization*, 2021. arXiv: 2104.02112 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2104.02112

[58] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, "Perceiver: General perception with iterative attention," in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 4651–4664.

[59] Y. Nie et al., "Towards question-answering as an automatic metric for evaluating the content quality of a summary," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 813–829, 2021.

[60] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu, "Roformer: Enhanced transformer with rotary position embedding," *arXiv preprint arXiv:2104.09864*, 2021.

[61] W. Zhao, S. Eger, J. Bjerva, I. Augenstein, and R. McDonald, "Qaeval: Evaluating factual consistency of summaries with question answering," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021, pp. 5245–5258.

[62] C. Zhu, Y. Liu, J. Mei, and M. Zeng, *Mediasum: A large-scale media interview dataset for dialogue summarization*, 2021. arXiv: 2103.06410 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2103.06410

[63] A. Chowdhery et al., *Palm: Scaling language modeling with pathways*, 2022. arXiv: 2204.02311 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2204.02311

[64] A. Gu, K. Goel, and C. Ré, *Efficiently modeling long sequences with structured state spaces*, 2022. arXiv: 2111.00396 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2111.00396

[65] F. D. Keles, P. M. Wijewardena, and C. Hegde, *On the computational complexity of self-attention*, 2022. arXiv: 2209.04881 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2209.04881

[66] B. Y. Lin et al., "Truthfulqa: Measuring how models mimic human falsehoods," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, Association for Computational Linguistics, 2022, pp. 3217–3252.

[67] U. Shaham et al., *Scrolls: Standardized comparison over long language sequences*, 2022. arXiv: 2201.03533 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2201.03533

[68] J. Li et al., "Blip-2: Bootstrapped language-image pretraining with frozen image encoders and large language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[69] B. Peng et al., *Rwkv: Reinventing rnns for the transformer era*, 2023. arXiv: 2305.13048 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2305.13048

[70] M. Poli et al., "Hyena hierarchy: Towards larger convolutional language models," *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

[71] C. Raffel et al., *Exploring the limits of transfer learning with a unified text-to-text transformer*, 2023. arXiv: 1910.10683 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1910.10683

[72] Y. Sun et al., *Retentive network: A successor to transformer for large language models*, 2023. arXiv: 2307.08621 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2307.08621

[73] H. Touvron et al., *Llama: Open and efficient foundation language models*, 2023. arXiv: 2302.13971 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2302.13971

[74] A. Vaswani et al., *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1706.03762

[75] Y. Bai et al., *Longbench: A bilingual, multitask benchmark for long context understanding*, 2024. arXiv: 2308.14508 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2308.14508

[76] F. L. Bronnec et al., *Locost: State-space models for long document abstractive summarization*, 2024. arXiv: 2401.17919 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2401.17919

[77] T. Dao and A. Gu, *Transformers are ssms: Generalized models and efficient algorithms through structured state space duality*, 2024. arXiv: 2405.21060 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2405.21060

[78] X. Dong et al., *Hymba: A hybrid-head architecture for small language models*, 2024. arXiv: 2411.13676 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2411.13676

[79] P. Glorioso et al., *The zamba2 suite: Technical report*, 2024. arXiv: 2411.15242 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2411.15242

[80] P. Glorioso et al., *Zamba: A compact 7b ssm hybrid model*, 2024. arXiv: 2405.16712 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2405.16712

[81] A. Gu and T. Dao, *Mamba: Linear-time sequence modeling with selective state spaces*, 2024. arXiv: 2312.00752 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2312.00752

[82] K. Hilsenbek, *Breaking the attention bottleneck*, 2024. arXiv: 2406.10906 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2406.10906

[83] S. Hwang, A. Lahoti, T. Dao, and A. Gu, *Hydra: Bidirectional state space models through generalized matrix mixers*, 2024. arXiv: 2407.09941 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2407.09941

[84] O. Lieber et al., *Jamba: A hybrid transformer-mamba language model*, 2024. arXiv: 2403.19887 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2403.19887

[85] T. C. Nauen, S. Palacio, and A. Dengel, "Taylorshift: Shifting the complexity of self-attention from squared to linear (and back) using taylor-softmax," in *Pattern Recognition*. Springer Nature Switzerland, Dec. 2024, pp. 1–16, ISBN: 9783031781728. DOI: 10.1007/978-3-031-78172-8_1 [Online]. Available: http://dx.doi.org/10.1007/978-3-031-78172-8_1

[86] S. Omranpour, G. Rabusseau, and R. Rabbany, *Higher order transformers: Efficient attention mechanism for tensor structured data*, 2024. arXiv: 2412.02919 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2412.02919

[87] Z. Shen et al., *Slimpajama-dc: Understanding data combinations for llm training*, 2024. arXiv: 2309.10818 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2309.10818

[88] R. Waleffe et al., *An empirical study of mamba-based language models*, 2024. arXiv: 2406.07887 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2406.07887

[89] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," *arXiv preprint arXiv:2401.09417*, 2024. arXiv: 2401.09417 [cs.CV].

[90] X. Chen et al., *Transmamba: Fast universal architecture adaption from transformers to mamba*, 2025. arXiv: 2502.15130 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2502.15130

[91] L. Ren, Y. Liu, Y. Lu, Y. Shen, C. Liang, and W. Chen, *Samba: Simple hybrid state space models for efficient unlimited context language modeling*, 2025. arXiv: 2406.07522 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2406.07522

[92] S. Somvanshi, M. M. Islam, M. S. Mimi, S. B. B. Polock, G. Chhetri, and S. Das, *From s4 to mamba: A comprehensive survey on structured state space models*, 2025. arXiv: 2503.18970 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2503.18970

[93] B. Wu, J. Shi, Y. Wu, N. Tang, and Y. Luo, *Transxssm: A hybrid transformer state space model with unified rotary position embedding*, 2025. arXiv: 2506.09507 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2506.09507

[94] Y. Li et al., *Transmamba: A sequence-level hybrid transformer-mamba language model*, 2026. arXiv: 2503.24067 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2503.24067