# POLITECNICO DI TORINO

*Department of Electronics and Telecommunications*

*Master's Degree in ICT for Smart Societies*


Politecnico
di Torino


## Enhancing a JSCC Architecture with Retransmission Based on Semantic Feedback

**Supervisors**

Prof. Carla Fabiana Chiasserini

Dr. Marco Palena

Eng. Roberto Fantini

Eng. Elisa Zimaglia

**Candidate**

Yasmin Awed Mohamud

July 2025

# Abstract

Traditional communication systems have long prioritized the accurate transmission of bit sequences, focusing more on symbol-level precision than on the actual meaning of the transmitted information. This approach, while effective in many technical scenarios, can be limiting when the goal is to ensure that the receiver understands the intended message rather than its exact form.

Semantic communication introduces a shift in perspective by aiming to convey meaning instead of replicating every transmitted symbol. Recent advancements in deep learning, particularly through Transformer-based architectures, have made it possible to develop systems that optimize for semantic understanding rather than syntactic accuracy.

This thesis proposes the enhancement of a semantic communication system based on a Joint Source–Channel Coding (JSCC) framework for text transmission, with several modifications aimed at improving its robustness and adaptability in real-world conditions. The proposed solution includes a feedback mechanism that enables selective retransmissions and leverages reference-free quality metrics to evaluate decoding success without needing a ground truth comparison.

In addition, the thesis presents an experimental evaluation of the system, carried out under different wireless channel conditions. The goal is not only to test its performance but also to observe how the feedback mechanism responds in each scenario, offering insights into its behavior. The results show the impact achieved with the proposed enhancements and provide hints towards further approaches for an effective semantic-based retransmission system.

# Table of Contents

# List of Figures

# List of Tables

# Acronyms

- **ARQ** – Automatic Repeat reQuest
- **BERTScore** – BERT-based Semantic Similarity Score
- **BLEU** – Bilingual Evaluation Understudy
- **BLEURT** – Bilingual Evaluation Understudy with Representations from Transformers
- **CDL-B** – Clustered Delay Line – B
- **CE** – Cross-Entropy
- **CIT** – Classic Information Theory
- **CoLA** – Corpus of Linguistic Acceptability
- **JSCC** – Joint Source–Channel Coding
- **DeepJSCC** – Deep Joint Source–Channel Coding
- **DeepSC** – Deep Semantic Communication
- **FAM** – Feedback Attention Memory
- **FDE** – Feature Distortion Evaluation
- **FIR** – Feature Importance Ranking
- **HARQ** – Hybrid Automatic Repeat reQuest
- **LiDAR** – Light Detection and Ranging
- **MCC** – Matthews Correlation Coefficient
- **METEOR** – Metric for Evaluation of Translation with Explicit ORdering
- **MI** – Mutual Information
- **MIMO** – Multiple Input Multiple Output

- **MLM** – Masked Language Modeling
- **MMSE** – Minimum Mean Squared Error
- **NLG** – Natural Language Generation
- **NLP** – Natural Language Processing
- **NNs** – Natural Networks
- **NSP** – Next Sentence Prediction
- **PPL** – Perplexity
- **PPLu** – Unigram-Normalized Perplexity
- **SFC** – Space Frequency Coding
- **SimCRC** – Similarity-based Cyclic Redundancy Check
- **SimHARQ-I** – Semantic HARQ with Chase Combining
- **SimHARQ-II** – Semantic HARQ with Incremental Redundancy
- **SNR** – Signal-to-Noise Ratio
- **SVD** – Singular Value Decomposition
- **ToM** – Theory of Mind
- **V2V** – Vehicle-to-Vehicle
- **ZF** – Zero-Forcing

# 1 Introduction

## 1.1 Background and Motivation

Classic Information Theory (CIT), introduced by Shannon in 1948, provides the mathematical foundations of modern communication systems. As presented in [26], the classic communication system consists of a source, encoder, channel (affected by noise), decoder, and destination, as illustrated in Figure 1. This framework focuses on the uncertainty of information and introduces four fundamental metrics used to evaluate communication performance: entropy, mutual information, channel capacity, and the rate-distortion function.

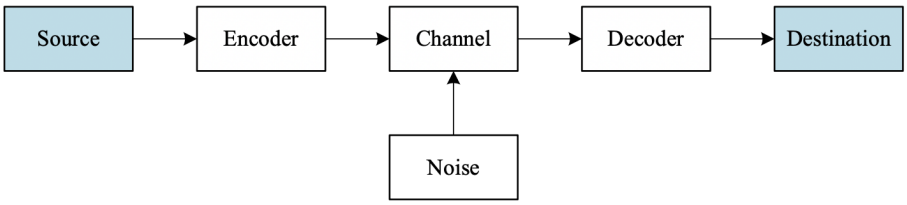



*Figure 1: Block diagram of a classical communication system. Source: [26].*

While this model has driven decades of research and development in telecommunications, it only addresses what Weaver defined as Level A of communication, which is concerned with the technical problem of transmitting symbols accurately. Shannon explicitly stated that the semantic aspects of communication are irrelevant to the engineering problem [26]. As a result, meaning, interpretation, and context are excluded from the classical model.

To overcome this limitation, recent works have proposed Semantic Communication as a natural extension of CIT. In this paradigm, the primary goal is not the faithful reconstruction of a bit sequence, but the correct interpretation of the intended meaning at the receiver side. This corresponds to Level B in Weaver's classification [37], where the emphasis is on semantics rather than syntax.

The semantic communication system introduced in [26] is shown in Figure 2. In this extended model, the source and destination each contain both a syntactic and a semantic component. Synonymous mapping functions are used to associate sets of syntactic elements with shared semantic content, enabling the system to operate on semantic equivalence classes rather than raw bit sequences. A similar demapping process is performed at the receiver.

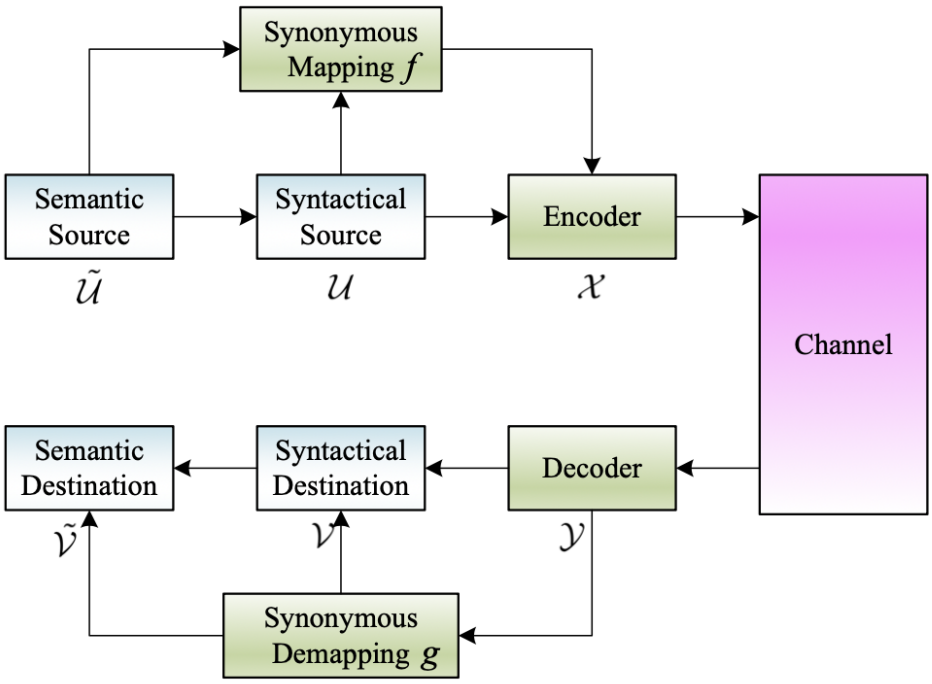



*Figure 2: Block diagram of a semantic communication system. Source: [26].*

This reformulation makes it possible to reduce the volume of transmitted information by focusing on the transmission of meaning. As a result, semantic communication has the potential to significantly improve communication efficiency, especially in scenarios with limited bandwidth or strict latency constraints. Moreover, it provides the theoretical foundation for intelligent, context-aware communication in future systems such as 6G.

This field is still relatively new, and the enhancements proposed in this thesis aim to increase the autonomy, reliability, and semantic robustness of DeepSC, a JSCC framework for text transmission, through adaptive evaluation, feedback-based retransmission, and memory-augmented decoding.

## 1.2 Research Direction

This thesis extends DeepSC by introducing a series of enhancements to make semantic communication more autonomous and robust. The proposed contributions are as follows:

- **Reference-free semantic evaluation:** A decision mechanism based on perplexity, unigram-normalized perplexity, and grammatical acceptability (via CoLA) is introduced, allowing the receiver to assess the quality of decoded messages without relying

on reference sentences.

- **Retransmission loop based on semantic quality:** A retransmission strategy is implemented in which the decoder triggers a new transmission when the decoded sentence is deemed semantically unacceptable. This process continues until a satisfactory output is obtained or a maximum number of attempts is reached.

- **Memory-augmented decoding:** During retransmission, the decoder receives both the current and previous received signals as input. This fused representation allows the decoder to incorporate information from prior attempts, enabling semantic refinement and improved output quality.

## 1.3 Objectives

The main objective of this work is to improve the practical viability of semantic communication in realistic, reference-free settings. Specifically, the work aims to:

1. Enable semantic retransmission decisions using reference-free evaluation metrics.

2. Integrate a decoder memory mechanism that conditions future predictions on previously received signals.

3. Design a feedback loop at inference time that incorporates semantic-aware retransmission policies.

4. Evaluate the system under both reference-based and reference-free conditions using established metrics such as BLEURT, BERTScore, PPL, PPLu, and CoLA.

## 1.4 Outline

The remainder of this document is structured as follows:

- **Section 2** reviews the state of the art in semantic communication. It first presents the DeepSC framework and its Transformer-based architecture, then surveys relevant research on evaluation metrics (both reference-based and reference-free), as well as recent extensions addressing feedback mechanisms, retransmission strategies, and memory-augmented decoding.

- **Section 3** describes the proposed architectural modifications and their implementation.
- **Section 4** reports experimental results and analysis.
- **Section 5** provides conclusions and future directions.

# 2 State of the Art

## 2.1 Main Research Directions in Semantic Communication

The semantic communication paradigm has been thoroughly analyzed in [22], which identifies three principal research directions: Theory of Mind (ToM), Generative AI, and Deep Joint Source-Channel Coding (DeepJSCC). These approaches aim to go beyond classical information theory by embedding semantic understanding into the entire communication process. Figure 3 illustrates their conceptual organization and mutual relationships.

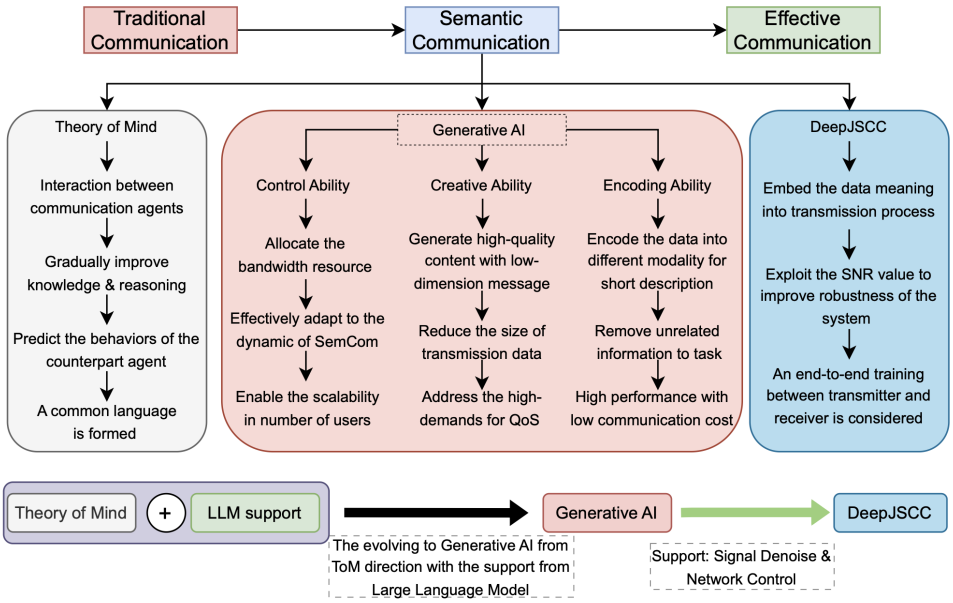



*Figure 3: Key directions in semantic communication: Theory of Mind, Generative AI, and DeepJSCC. Source: [22].*

**Theory of Mind (ToM)** In the Theory of Mind (ToM) approach, agents construct internal models of each other's knowledge, beliefs, and intentions. This cognitive modeling enables more adaptive and efficient communication, as messages are generated not just for correctness but for interpretability from the recipient's perspective.

An example of this approach is proposed in the pragmatic semantic communication system by Thomas et al. [33], which introduces a two-level feedback architecture grounded in ToM reasoning. At the first level (Level A), physical transmission is adapted to the channel state through traditional feedback mechanisms such as SNR or CQI. At the second level,

semantic feedback is integrated using a cognitive model of the receiver's neural networks (NNs), allowing the transmitter to infer how the receiver might interpret or misinterpret a given message.

The system includes a semantic encoder and decoder that are enhanced with ToM modules capable of modeling the neural behavior of the communication counterpart. These models are used to simulate and anticipate semantic ambiguity, improving both message selection and refinement. The overall architecture is illustrated in Figure 4.

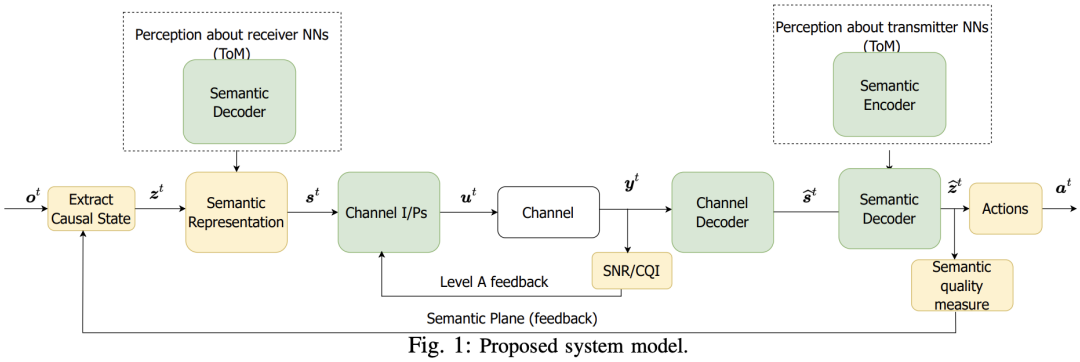


Fig. 1: Proposed system model.

*Figure 4: Pragmatic semantic communication framework with dual-level feedback and Theory of Mind modeling. Source: [33].*

A more comprehensive cognitive integration is seen in the MindForge framework by Colle et al. [25], which equips embodied agents with a full causal ToM template for continual and collaborative learning. MindForge is designed to operate in complex, open-ended environments such as Minecraft, where agents must communicate, plan, and adapt over long time horizons.

The system incorporates a structured model of cognition including percepts, beliefs, desires, and actions, all grounded in a dynamic task environment. It uses episodic, semantic, and procedural memory to track prior experience and align current behavior with inferred partner goals. Communication is mediated by a dedicated module that interacts with a critic and execution planner, enabling agents to plan context-aware interactions and learn from multi-agent dialogue.

The MindForge architecture is shown in Figure 5, where the Causal ToM Template operates in coordination with a memory system and task-specific modules. Belief inference is used not only for action selection but also for dialogue planning and policy adaptation.

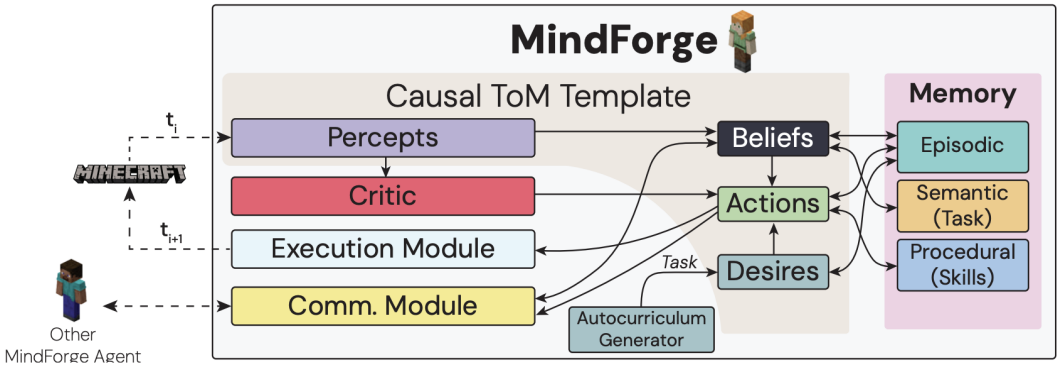



*Figure 5: MindForge framework: integration of causal Theory of Mind with episodic, semantic, and procedural memory modules in a multi-agent environment. Source: [25].*

Both frameworks exemplify the growing role of ToM in semantic communication systems. By reasoning about how others interpret and act upon information, agents can dynamically adjust their encoding, select context-relevant utterances, and improve communication efficiency in noisy, ambiguous, or multi-agent scenarios.

**Generative AI-Based Semantic Communication** Generative AI facilitates semantic communication by allowing the transmitter to abstract high-dimensional sensory or linguistic inputs into compact, meaning-preserving latent representations. These can then be decoded or regenerated by the receiver, yielding semantically equivalent reconstructions even under limited bandwidth or noisy channels.

A comprehensive overview of this vision is presented by Grassucci et al.[5], who contrast traditional reconstruction-oriented communication systems with semantic-preserving generative approaches. Instead of transmitting raw or heavily encoded signals, the transmitter extracts scene-level semantics (e.g., textual descriptions or sparse semantic maps) and sends these compressed representations. The receiver uses a generative model to reconstruct a plausible version of the original scene. As illustrated in Figure 6, this approach significantly reduces bandwidth demands while preserving communicative intent.

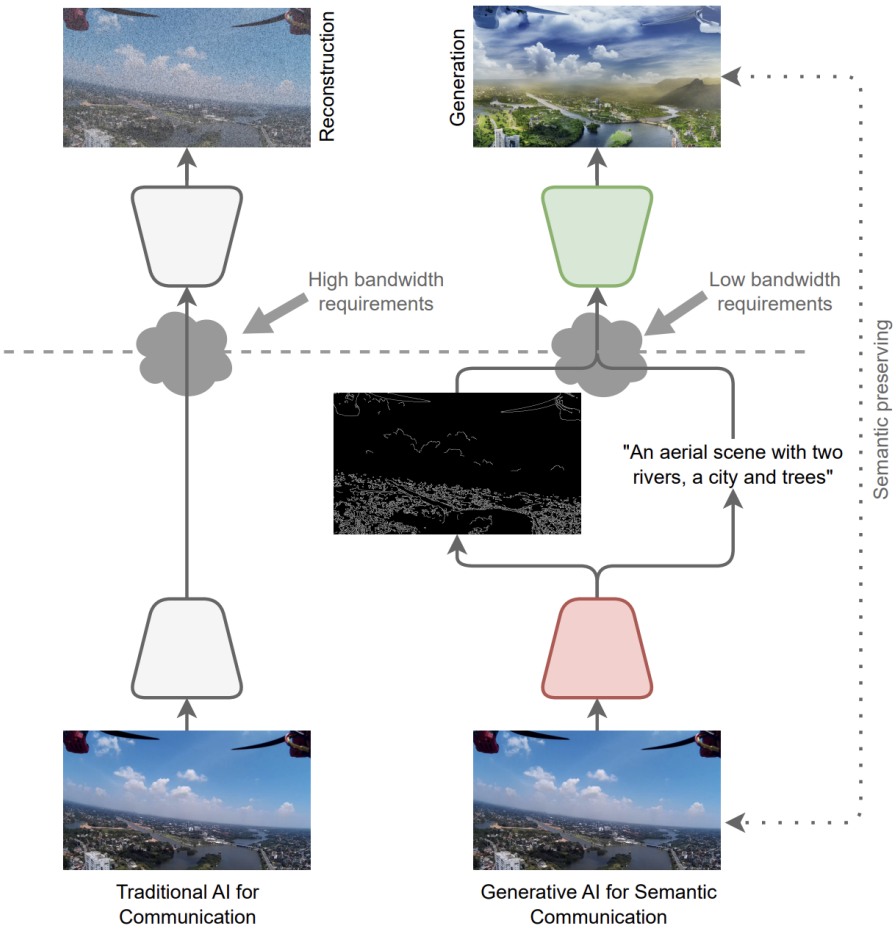



*Figure 6: Comparison between traditional AI-based reconstruction and generative AI-based semantic communication. The latter preserves meaning while reducing bandwidth. Source: [5].*

A more task-specific implementation is proposed by Zhang et al.[19], who design a semantic communication system that progressively refines message quality through generative inference. The system, shown in Figure 7, relies on a Swin-Transformer encoder to extract patch-level semantic embeddings, which are then decoded at the receiver via a diffusion model. This combination enables robustness in low-SNR environments and supports multi-user settings by distributing semantic features across users with distinct refinement stages.

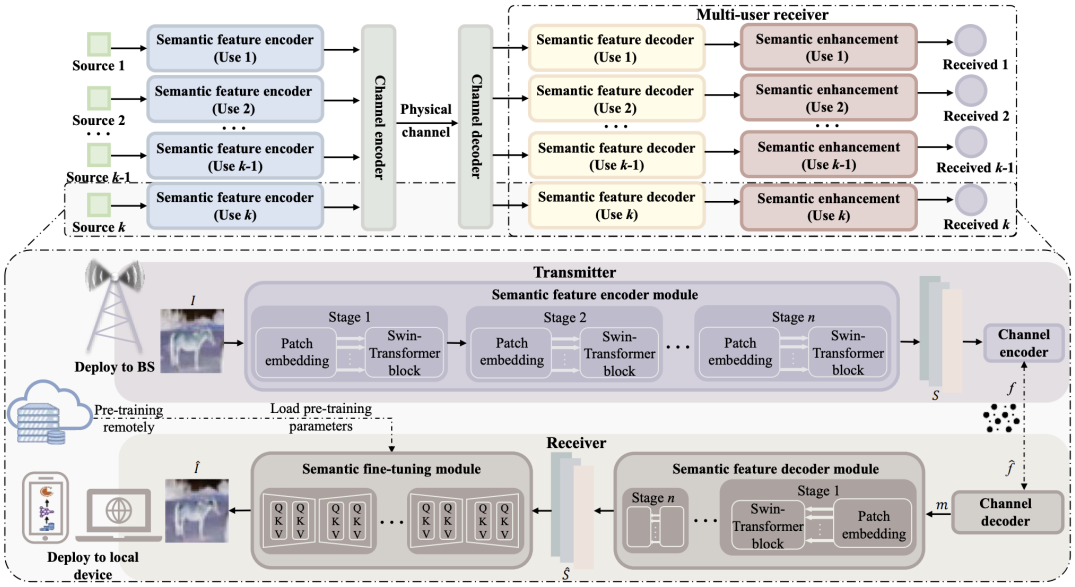



*Figure 7: Semantic successive refinement architecture with Swin Transformer encoder and diffusion-based decoder across multiple users. Source: [19].*

The foundation of these systems rests on several families of generative models, each with distinct strengths. As summarized in Figure 8, Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), Diffusion Models (DMs), and Transformer-based architectures offer different trade-offs in sample quality, training stability, and representation flexibility. Diffusion models, in particular, have gained popularity in semantic communication due to their robustness in noisy environments and their capacity to model uncertainty during reconstruction.

| Technologies | Architectures | Features |
|---|---|---|
| **Generative Adversarial Network (GAN)** | • *Generator Network*: Generate data that closely resembles real data.<br>• *Discriminator Network*: Evaluate the authenticity of the generated data and engages in adversarial training with the generator.<br>∗ **Variants**: *Conditional GAN*, *Wasserstein GAN*, *StyleGAN* | Generate high-quality images, but training is unstable. |
| **Variational Autoencoder (VAE)** | • *Encoder Network*: Map input to a latent space distribution.<br>• *Decoder Network*: Reconstruct the data from samples in the latent space to approximate the original input.<br>∗ **Variants**: *Vector Quantized VAE*, *Variational Auto-Encoding GAN* | Stabilize training, but generate images with lower sample quality. |
| **Diffusion Model (DM)** | • *Forward Process*: Gradually add noise to the original data in a step-by-step process.<br>• *Denoising Process*: Revert the noise from the normal distribution.<br>∗ **Variants**: *Stable Diffusion*, *Denoising Diffusion Probabilistic Model* | Generate high-quality images, with stable training. |
| **Transformer-based Model** | • *Self-Attention*: Learns the dependencies between elements in the input by computing attention weights.<br>∗ **Variants**: *Generative Pre-trained Transformer (GPT)* | Capable of handling long-range dependencies parallelizable training. |

*Figure 8: Comparison of popular generative architectures for semantic communication. Each family offers different strengths for encoding and reconstruction. Source: [16].*

In the context of vehicular communication, Lu et al.[16] propose the GMSC (Generative

AI-enhanced Multi-modal Semantic Communication) system. Their design addresses the real-time and bandwidth-constrained requirements of Internet of Vehicles (IoV) by fusing data from multiple sensors (e.g., camera, LiDAR, GPS), encoding it semantically, and applying diffusion models for robust reconstruction and reasoning. Figure 9 shows the dual pipeline for digital and analog semantic transmission, where tasks such as scene understanding and BEV (bird's-eye view) prediction are supported through generative decoding.

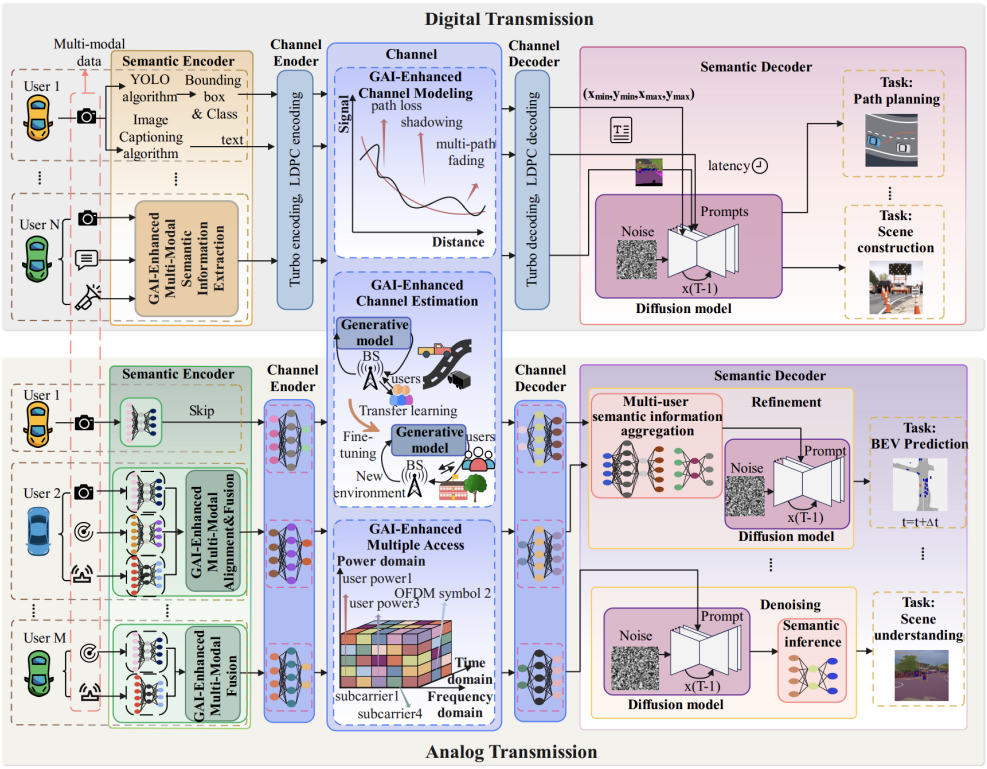



*Figure 9: GMSC architecture for multi-modal semantic communication in vehicular systems using generative AI. The framework includes both digital and analog pipelines. Source: [16].*

These examples illustrate how generative AI enables communication systems to prioritize meaning over syntax, leading to more flexible, robust, and bandwidth-efficient designs. By exploiting the generative prior learned from massive datasets, these models bridge the gap between compression, reasoning, and reconstruction in semantic communication.

**Deep Joint Source–Channel Coding (DeepJSCC)** DeepJSCC unifies source and channel coding into a single neural framework that learns to transmit semantic information directly over noisy wireless channels. This joint optimization enables robust communication with reduced latency and better semantic preservation compared to traditional separation-based

designs. This thesis work falls within this research direction, focusing on neural joint source–channel coding and the integration of semantic feedback and retransmission strategies.

A representative example is MambaJSCC, proposed by Wu et al. [35]. This model integrates visual state-space modeling into a DeepJSCC framework, enabling real-time adaptability through channel-aware encoding. The encoder captures temporal and spatial features using patch-based 2D embedding and VSSM-CA blocks, while Channel State Information (CSI) is encoded to guide both the encoder and decoder during transmission. The complete pipeline is illustrated in Figure 10.

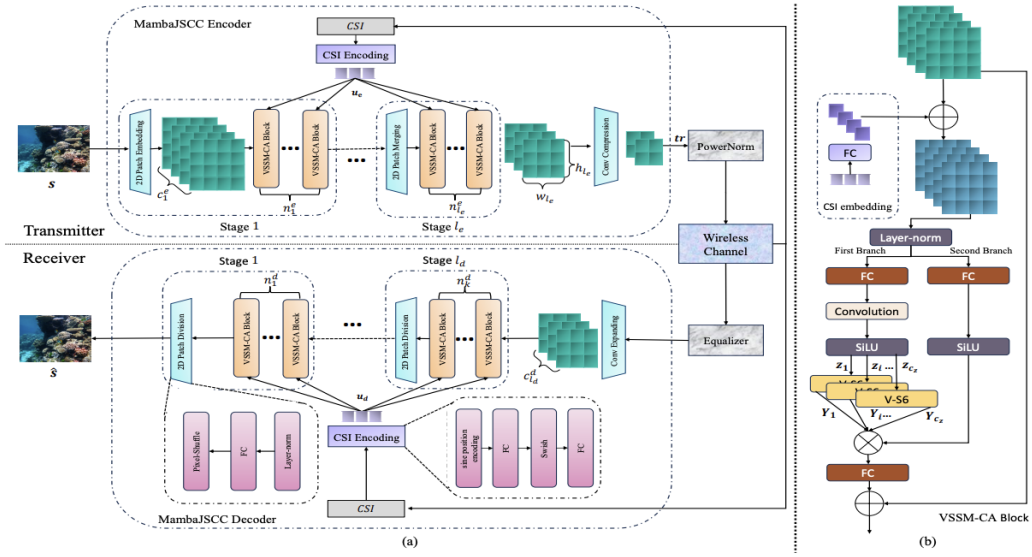



*Figure 10: MambaJSCC architecture incorporating CSI-guided visual state-space modeling for semantic image transmission. Source: [35].*

A more modular design is offered by Huang et al. [15], who propose **D$^2$-JSCC**, a digital variant of DeepJSCC. Their approach combines deep entropy coding with digital channel encoding and supports bit-level control while preserving semantic content. This structure enables joint optimization over compression and reliability metrics. As shown in Figure 11, their system bridges neural compression with traditional channel coding strategies, benefiting from digital flexibility.

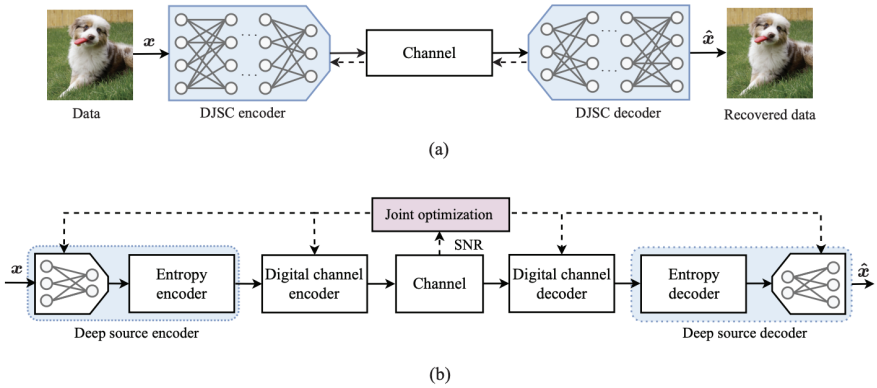
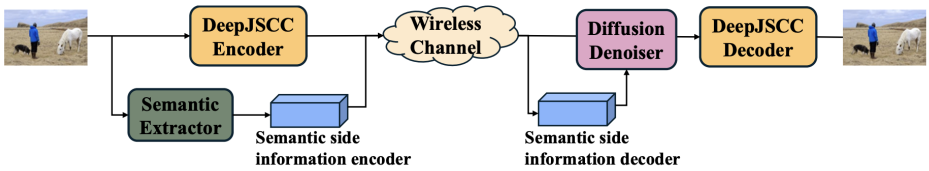



*Figure 11: (a) End-to-end DeepJSCC encoder-decoder pipeline. (b) Modular $D^2$-JSCC architecture with entropy coding and joint optimization. Source: [15].*

To improve semantic robustness in adverse channel conditions, Zhu et al. [23] propose **SGD-JSCC**, which augments DeepJSCC with a diffusion denoising module. Their model introduces semantic side information to guide the generative refinement process. As shown in Figure 12, the decoder leverages diffusion-based generation to reconstruct semantically meaningful content, compensating for degraded latent features.


(a) Standard DeepJSCC scheme.




(b) Proposed semantics guided DeepJSCC scheme.

*Figure 12: Standard vs. semantics-guided DeepJSCC with diffusion denoiser and auxiliary side information. Source: [23].*

In task-oriented scenarios, Park et al. [30] design a KL-regularized DeepJSCC framework aimed at stabilizing semantic representations during transmission. Their system aligns the posterior of received latent variables with the prior distribution learned during training, using KL-divergence regularization. This setup is well-suited for applications like image captioning

or classification, as depicted in Figure 13, where the decoder directly infers symbolic outputs from noisy signals.

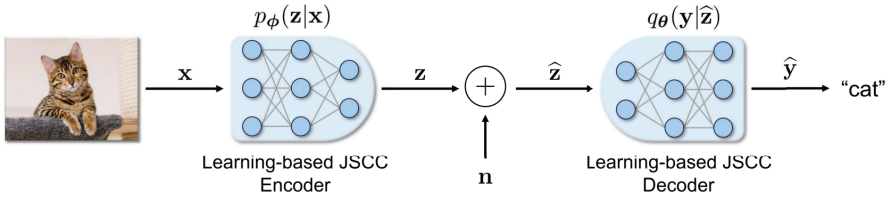



*Figure 13: KL-regularized DeepJSCC for robust task-oriented semantic communication. The decoder predicts symbolic labels directly from noisy features. Source: [30].*

Collectively, these innovations demonstrate the flexibility of DeepJSCC in combining end-to-end differentiability, semantic robustness, and cross-layer optimization. By adopting generative models, attention mechanisms, and hybrid digital-analog designs, modern DeepJSCC systems push the boundaries of what semantic communication can achieve under realistic wireless constraints.

**Interrelation and Evolution** These three directions are not isolated. Theory of Mind can evolve into Generative AI-based models by integrating external knowledge and neural priors. Generative architectures, in turn, benefit from DeepJSCC techniques to ensure semantic reliability over real-world wireless channels. Together, they form a unified foundation for intelligent, adaptive, and efficient communication systems.

## 2.2 DeepSC: A Transformer-Based Semantic Communication System

DeepSC (Deep Learning Enabled Semantic Communication) is an architecture that implements a Joint Source-Channel Coding (JSCC) strategy using deep learning, with a specific focus on preserving the semantic meaning of transmitted messages. Unlike traditional communication systems that treat source coding (e.g., compression) and channel coding (e.g., error correction) as separate blocks, JSCC integrates these functions into a single end-to-end model. DeepSC further specializes this by ensuring that the transmitted information maintains its semantic integrity, even if the exact symbol sequence is altered due to channel imperfections.

The original DeepSC model, introduced by Xie et al. [10], was designed for text-based communication using a Transformer-based encoder-decoder architecture. This approach allows the system to extract high-level semantic features from natural language text and transmit them efficiently over noisy wireless channels. The model was later extended by Grieco [7] to better simulate real-world conditions and support benchmarking against standard 5G New Radio (NR) communication systems. These improvements include support for realistic physical channels, Multiple-Input Multiple-Output (MIMO) transmission, and advanced equalization techniques, while preserving the interpretability and semantic fidelity of the original design.

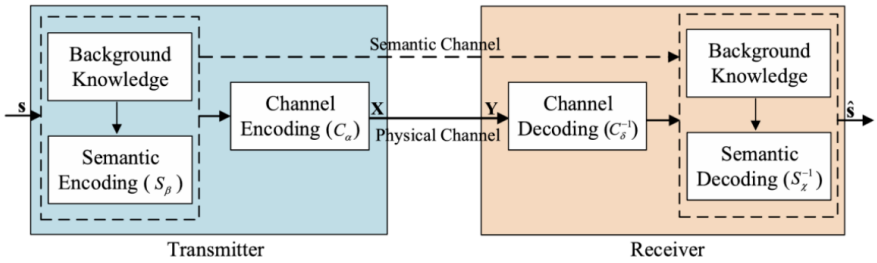



*Figure 14: General DeepSC semantic transceiver model. Source: [10].*

The transmitter maps the input sentence $\mathbf{s} = [w_1, w_2, \ldots, w_L]$ into a semantic representation via a transformer encoder $S_\beta$, followed by a channel encoder $C_\alpha$:

$$\mathbf{x} = C_\alpha(S_\beta(\mathbf{s})) \tag{1}$$

The signal is transmitted over a realistic physical channel with additive noise and MIMO fading:

$$\mathbf{y} = \mathbf{H} \cdot \mathbf{x} + \mathbf{n} \tag{2}$$

At the receiver, decoding is performed via:

$$\hat{\mathbf{s}} = S_\chi^{-1}(C_\delta^{-1}(\mathbf{y})) \tag{3}$$

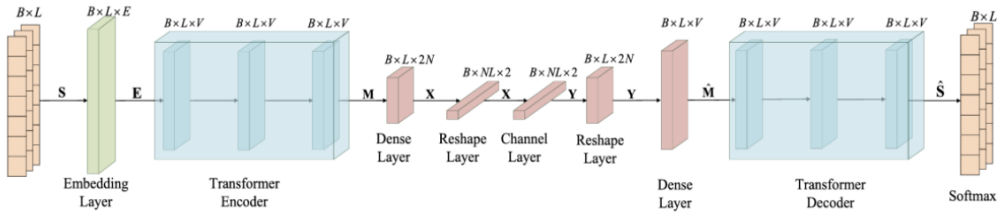



*Figure 15: Detailed implementation of the final DeepSC transformer-based model. Source: [10].*

The detailed architecture in Figure 15 illustrates the end-to-end transceiver pipeline using Transformer-based components. The input batch $\mathbf{S}$ has shape $B \times L$, where $B$ is the batch size and $L$ is the sentence length. Each word is embedded into a vector of size $E$, resulting in a tensor of shape $B \times L \times E$.

The key processing steps are:

- **Embedding Layer**: maps tokens to dense vectors $\mathbf{E} \in \mathbb{R}^{B \times L \times E}$.
- **Transformer Encoder**: extracts semantic features, producing $\mathbf{M} \in \mathbb{R}^{B \times L \times V}$, where $V$ is the latent dimension of the semantic space.
- **Dense Layer & Channel Encoder**: the semantic features are projected and reshaped into $\mathbf{X} \in \mathbb{R}^{B \times NL \times 2}$, representing complex-valued channel symbols.
- **Channel Layer**: simulates transmission through a realistic physical channel (AWGN, Rayleigh, Rician, or CDL-B MIMO).
- **Channel Decoder & Dense Layer**: reconstructs the semantic embedding $\hat{\mathbf{M}} \in \mathbb{R}^{B \times L \times V}$ from the received signal.
- **Transformer Decoder + Softmax**: estimates the original sentence $\hat{\mathbf{S}} \in \mathbb{R}^{B \times L}$.

The architecture incorporates advanced physical layer components:

- **CDL-B channel model** (3GPP-compliant), simulating multipath fading via clustered delay profiles.
- **MIMO transmission** (up to $32 \times 4$) using Space Frequency Coding (SFC) for diversity and robustness.

- **Pre-coding via Singular Value Decomposition (SVD)**:

$$\mathbf{H} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^H, \quad \mathbf{X}_{\text{tx}} = \mathbf{V} \cdot \mathbf{X}_{\text{MIMO}} \tag{4}$$

- **Equalization techniques**, including:
  - *Zero-Forcing (ZF)*:

$$\mathbf{X}_{\text{MIMO}} \approx \mathbf{U}^H\boldsymbol{\Sigma}^H\mathbf{Y} \tag{5}$$

  - *Minimum Mean Squared Error (MMSE)*:

$$\mathbf{W}_{\text{MMSE}} = \mathbf{H}_{\text{eq}}^H \left(\mathbf{H}_{\text{eq}}\mathbf{H}_{\text{eq}}^H + N_0\mathbf{I}\right)^{-1} \tag{6}$$

The system is trained end-to-end using a composite loss function that combines semantic accuracy and channel robustness:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}}(\mathbf{s}, \hat{\mathbf{s}}) - \lambda\mathcal{L}_{\text{MI}}(\mathbf{x}, \mathbf{y}) \tag{7}$$

where the mutual information loss is computed using a neural estimator $f_T$:

$$\mathcal{L}_{\text{MI}} = \mathbb{E}_{p(x,y)}[f_T] - \log\left(\mathbb{E}_{p(x)p(y)}[e^{f_T}]\right) \tag{8}$$

The training process is performed in two stages, as illustrated in Figure 16. First, the mutual information estimator $f_T$ is trained. Then, the full transceiver is optimized via stochastic gradient descent using the total loss function.

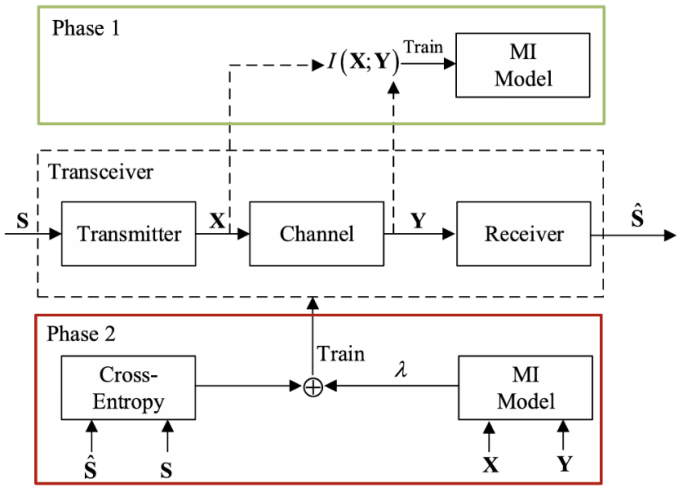



*Figure 16: Two-phase training process used in DeepSC. Phase 1 estimates the mutual information $I(\mathbf{X};\mathbf{Y})$, and Phase 2 performs end-to-end optimization using a combined loss function based on cross-entropy and mutual information. Source: [7].*

These architectural extensions enable DeepSC to be evaluated under real-world communication conditions, making it a suitable framework for benchmarking semantic communication against traditional systems such as 5G NR.

## 2.3 ARQ and HARQ Mechanisms in DeepSC

Automatic Repeat reQuest (ARQ) and Hybrid Automatic Repeat reQuest (HARQ) are classical error control mechanisms used in communication systems to ensure data reliability over unreliable or noisy channels. In ARQ protocols, the receiver detects transmission errors and requests retransmission of corrupted data packets via acknowledgment (ACK) and negative acknowledgment (NACK) messages. HARQ extends this principle by combining retransmissions with error-correction coding, allowing the receiver to incrementally decode messages using both newly received and previously stored data.

While traditional ARQ and HARQ operate at the bit or symbol level, their reinterpretation in the context of semantic communication allows for new opportunities. Here, the goal is not just to correct corrupted bits, but to preserve the intended meaning of the transmitted content. This shift requires feedback mechanisms that assess semantic fidelity and trigger retransmissions only when the degradation affects comprehension or task performance. The integration of such semantic-aware ARQ/HARQ strategies into DeepSC architectures opens

up new avenues for improving communication efficiency, robustness, and adaptability.

Recent research has proposed several frameworks that extend ARQ and HARQ principles to operate on semantic representations. The most notable among these include SCAN, SemHARQ, and SimHARQ, which introduce feedback-driven mechanisms that consider content relevance, task utility, and semantic distortion during retransmission decisions. In the following, these approaches are reviewed and analyzed for their integration potential into DeepSC systems.

**SCAN: Semantic Communication with Adaptive Feedback** In [8], the authors propose the SCAN (Semantic Communication with Adaptive chaNnel feedback) framework, which reinterprets the classical ARQ paradigm through a semantic lens. Instead of relying solely on bit-level retransmission, SCAN introduces a content-aware mechanism that adapts both the transmission and feedback strategy based on semantic relevance and channel dynamics.

At the core of SCAN is the concept of *Semantic Distortion Outage Probability (SDOP)*, a learned function that estimates the probability of exceeding an acceptable semantic distortion threshold under given channel conditions. The receiver evaluates this probability based on the received signal $\mathbf{y}$, estimated channel state information (CSI), and the intended task (e.g., image classification or captioning). This SDOP value is sent back to the transmitter through a dedicated feedback channel, guiding the selection of compression levels and modulation parameters in future transmissions.

SCAN's feedback loop is shown in Figure 17. The receiver side includes a CSI encoder that compresses the estimated channel matrix and semantic quality signals into a compact feedback vector $\mathbf{z}_h$, which is transmitted to the sender. The transmitter then decodes $\mathbf{z}_h$ and, using a learned performance evaluation module, dynamically selects one of multiple DeepSC encoder branches optimized for different semantic complexities. These branches vary in compression strength, trading off fidelity and bandwidth. A shared MIMO precoding module ensures that the representation $\mathbf{z}$ is efficiently adapted to the channel matrix $\mathbf{H}$.

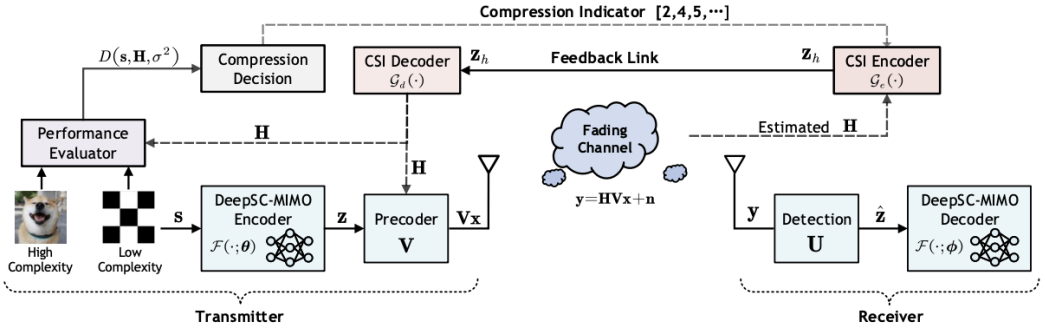



*Figure 17: SCAN system architecture. The receiver estimates the semantic distortion and channel state, encodes them into a feedback vector, and the transmitter dynamically adjusts compression and coding. Source: [8].*

Another key innovation in SCAN is the use of deep reinforcement learning (DRL) to jointly optimize compression selection and retransmission strategies. The transmitter acts as a policy agent, observing the SDOP feedback and selecting compression indicators (e.g., {2, 4, 5, ...}) that correspond to encoder branches with different semantic abstraction levels. The DRL policy is trained to minimize semantic distortion while accounting for channel variability and transmission costs. Importantly, retransmissions are triggered selectively and can focus on semantically high-impact regions (e.g., facial features in an image), rather than indiscriminately repeating entire messages.

Furthermore, SCAN supports multi-level semantic abstraction, where semantic units are structured hierarchically. This enables the system to degrade gracefully under poor channel conditions by discarding less relevant semantic layers while preserving core meaning. Such a design ensures robustness and scalability for tasks requiring semantic generalization, such as object recognition, image captioning, or scene understanding.

In conclusion, SCAN represents a paradigm shift in the use of feedback for semantic communication—transforming it from a mere error detection mechanism into a tool for semantic relevance evaluation. Its adaptive design offers a promising foundation for extending DeepSC architectures toward more context-aware and resilient communication strategies.

**SemHARQ: Task-Oriented Semantic HARQ** A more recent contribution is presented in [14], which introduces *SemHARQ*, a semantic-aware Hybrid Automatic Repeat reQuest protocol designed specifically for multi-task semantic communication systems. In contrast

to conventional HARQ strategies that operate at the bit or packet level, SemHARQ performs retransmission selectively at the semantic feature level, guided by task-oriented quality assessments.

The overall architecture, illustrated in Figure 18, comprises several interconnected modules. At the transmitter side, a multi-task semantic encoder first extracts semantically meaningful representations $\mathbf{s}$ from the input data (e.g., images). These are then passed through a joint source–channel (JSC) encoder to produce a compact feature vector $\mathbf{f}$. The core novelty lies in the subsequent *Scalable Feature Selector for HARQ*, which prioritizes semantic features based on their importance for each downstream task. This module outputs a subset of features $\tilde{\mathbf{z}}_j$ to be normalized and transmitted over the wireless channel.

At the receiver, a *Retransmission Identification Module* compares the received features $\hat{\mathbf{z}}_j$ against expected semantic representations using a feature distortion evaluation (FDE) network. The result is a binary feedback vector $p_{j+1}$, which acts as a bitmask indicating which feature groups should be retransmitted. This mask is returned to the transmitter via the feedback channel in the form of a semantic-aware NAK signal.

A key component of SemHARQ is the *Feature Importance Ranking* (FIR) mechanism, which enables the system to adapt its retransmission strategy dynamically. Features that are considered semantically critical, either due to their relevance to the target task or based on learned importance weights, are prioritized for retransmission. This approach maximizes the utility of each additional transmission while operating under bandwidth constraints.

The decoder at the receiver side performs two stages: (i) decoding of the semantic features via the JSC-decoder, and (ii) execution of parallel downstream tasks, such as classification or identification, using task-specific performers. The architecture supports multiple concurrent tasks (e.g., object type, color), and is optimized end-to-end to minimize semantic distortion while maximizing task-specific accuracy.

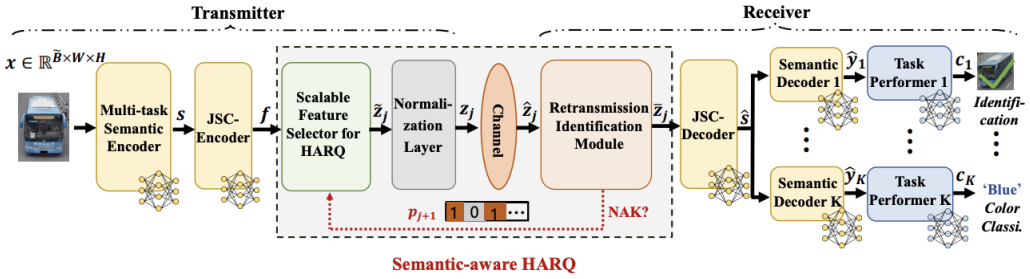



*Figure 18: SemHARQ system architecture: semantic features are prioritized and selectively retransmitted based on distortion and task relevance, with multi-task decoding at the receiver. Adapted from [14].*

To validate the performance of SemHARQ in realistic settings, the authors evaluate the system on the VeRi-776 dataset, which contains over 50,000 annotated images of 776 vehicles under varying camera viewpoints and lighting conditions. Each image is labeled with object ID, color, and vehicle type, enabling multi-task classification.

SemHARQ demonstrates substantial gains under low-SNR conditions. For instance, at an SNR of $-2$ dB, it achieves a rank-1 accuracy of 71.22% in vehicle re-identification, significantly surpassing both traditional HARQ and state-of-the-art semantic-aware HARQ baselines by over 40 percentage points. Comparable improvements are observed in auxiliary tasks such as color and type classification, with performance gains exceeding 10%.

These results confirm that SemHARQ's selective retransmission strategy, guided by semantic importance and distortion, offers a robust and bandwidth-efficient solution for multi-task semantic communication systems deployed in noisy wireless environments.

The modularity of SemHARQ allows seamless integration with scalable encoder architectures and pre-trained vision backbones, making it a robust candidate for real-world multi-modal semantic communication systems. By moving retransmission logic from the bit level to the semantic level, it lays the groundwork for more intelligent and context-aware feedback mechanisms in future networks.

**SimHARQ: Semantic HARQ for Cooperative Perception** In the domain of cooperative perception, [38] proposes a semantic communication framework specifically designed for vehicle-to-vehicle (V2V) transmission of LiDAR-derived features. Unlike conventional perception sharing, which often transmits raw or lightly processed point clouds, this system

adopts a semantic-first philosophy that prioritizes perceptual importance and task-specific relevance.

At the heart of the framework lies the use of *importance maps*, which quantify the semantic relevance of spatial regions within LiDAR data. These maps are generated through task-driven neural encoders and are used to rank the importance of feature segments prior to transmission. This ranking guides the allocation of communication resources, ensuring that high-value semantic components are prioritized under limited bandwidth.

To handle channel-induced errors, the authors introduce *SimCRC* (Similarity-based Cyclic Redundancy Check), a semantic-aware error detection module. Instead of relying solely on bit-level checksums, SimCRC computes the cosine similarity between received features and reference semantic embeddings. A semantic distortion threshold $\delta$ is used to determine whether retransmission is required. This thresholding mechanism provides more nuanced error detection by considering perceptual semantics rather than just raw accuracy.

Building on SimCRC, two variants of HARQ are introduced:

- **SimHARQ-I**: based on chase combining, this strategy retransmits identical copies of corrupted features and uses combining techniques at the receiver to improve robustness.
- **SimHARQ-II**: employs incremental redundancy by sending additional parity or semantic-dense information, progressively refining the received representation during retransmissions.

These mechanisms are embedded into a cooperative perception pipeline that fuses locally observed and received features for robust object detection. As illustrated in Figure 19, the pipeline includes modules for semantic encoding, importance assessment, SimCRC-based feedback generation, and progressive semantic fusion on the receiver side.

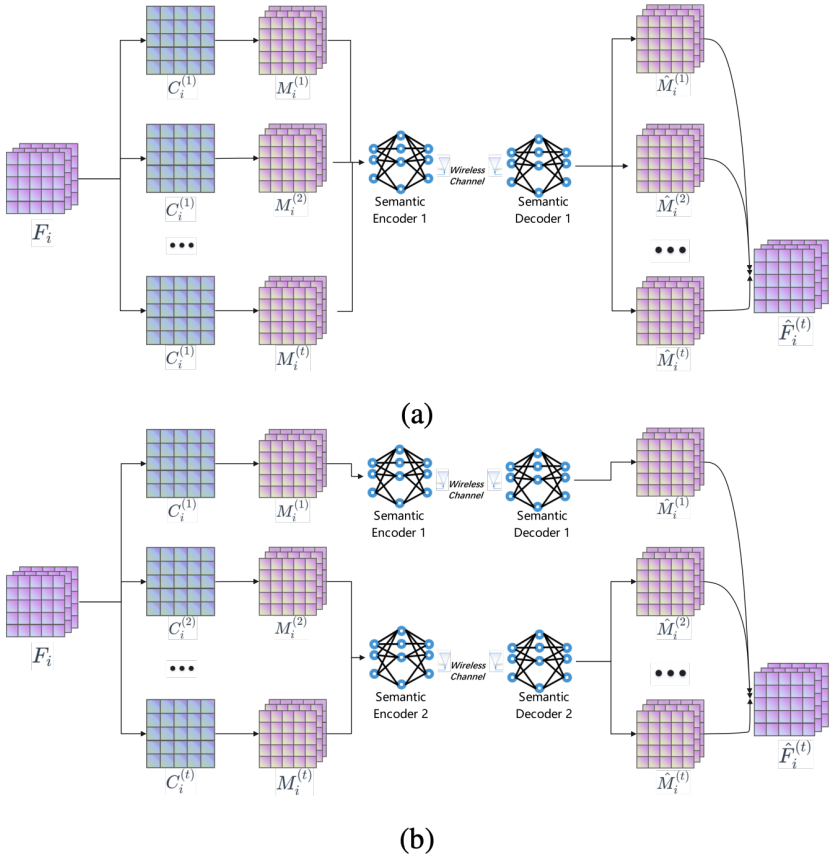



*Figure 19: SimHARQ framework for cooperative LiDAR feature sharing. Importance maps guide prioritized transmission, while semantic-aware HARQ mechanisms ensure robust perception under channel distortion. Adapted from [38].*

Extensive evaluations on the KITTI dataset demonstrate the effectiveness of the framework. Compared to traditional HARQ and baseline semantic communication systems, SimHARQ-II achieves up to 19% improvement in 3D object detection mAP under low-SNR conditions. Additionally, its ability to reduce unnecessary retransmissions results in up to 25% bandwidth savings without compromising detection accuracy.

Overall, SimHARQ showcases how semantic feedback, importance-driven prioritization, and incremental refinement can be synergistically combined to enable robust and bandwidth-efficient cooperative perception in vehicular networks. Its modular design also makes it amenable to integration with DeepSC-style architectures and future 6G V2X systems.

**Towards Feedback-Enhanced DeepSC** Building on the ideas presented in SCAN, SemHARQ, and SimHARQ, recent studies suggest that DeepJSCC-based architectures could benefit from semantic-aware ARQ/HARQ integration. For instance, semantic quality estimators such as

embedding similarity or task-specific classification scores can be used to trigger feedback responses. Upon detecting insufficient semantic reliability, the receiver may issue a NACK signal, leading to partial or full retransmissions. Some proposals explore token-level retransmission, conditional refinement through semantic encoders, or modulation of redundancy based on semantic importance [8, 14, 38].

Although implementations remain at the early stage, these directions highlight the potential of feedback mechanisms to enhance DeepSC robustness and adaptability, particularly in low-SNR and multi-task settings.

## 2.4 Memory-Augmented Feedback Loop in DeepSC

Recent advances in semantic communication have explored the integration of memory mechanisms into transceiver architectures to improve semantic continuity and contextual coherence. Memory modules enable the receiver to retain prior contextual knowledge, which is particularly relevant in sequential tasks such as instruction following, dialogue systems, or collaborative decision-making. A memory-augmented feedback mechanism enables the receiver to accumulate semantic knowledge over time and use it to refine message interpretation and guide adaptive retransmission strategies.

In [11], the authors propose a semantic communication system that incorporates an explicit memory module on the receiver side. The system operates in two distinct phases. In the first phase, referred to as *memory shaping*, contextual information such as prior dialogue turns or environmental observations is transmitted and stored in the receiver's memory. This data is not decoded directly but rather used to build semantic context. In the second phase, known as *task execution*, a new semantic message (for example, a question) is transmitted, and the receiver leverages both the received signal and its memory to reconstruct a semantically consistent output. This two-stage mechanism is illustrated in Figure 20, where subfigure (a) represents the memory shaping process and subfigure (b) shows the decoding process enriched by memory context.

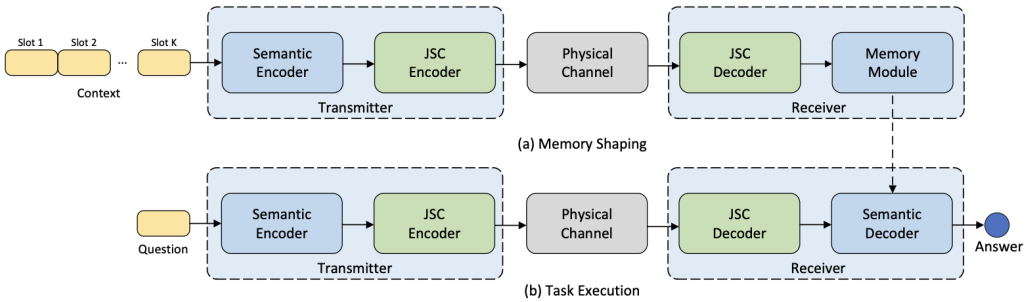



*Figure 20: Memory-augmented semantic communication system. (a) Memory shaping phase; (b) Task execution phase with memory-assisted decoding. Adapted from [11].*

An alternative design is presented in [4], where the authors introduce a Feedback Attention Memory (FAM) mechanism embedded within the Transformer architecture. Instead of maintaining a separate memory buffer, each token representation is recurrently propagated and selectively updated through successive transformer layers. As shown in Figure 21, this copy-and-replace mechanism allows long-range dependencies to be modeled efficiently, enabling contextual refinement without significantly increasing model complexity.

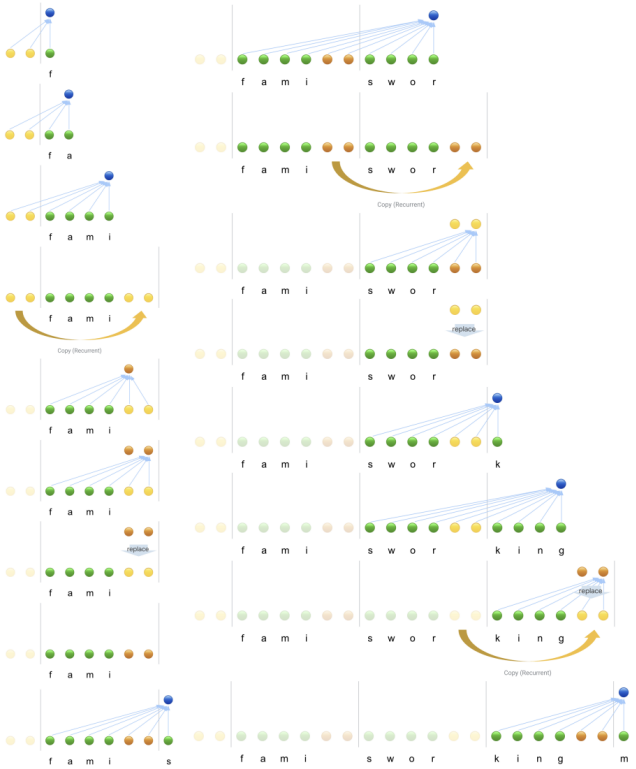


*Figure 21: Feedback Attention Memory (FAM) mechanism in TransformerFAM. Token representations are recurrently copied and selectively updated across transformer blocks. Adapted from [4].*

In addition to explicit memory mechanisms, the concept of implicit memory has been explored in [27]. The Implicit Memory Transformer relies on left-context attention to reference previously computed activations without the need for an external memory buffer. This approach enables the model to maintain semantic continuity across sequential inputs with reduced computational overhead, which is particularly valuable in real-time or low-latency semantic communication scenarios.

Explicit memory shaping [11], internal memory updating through attention recurrence [4], and implicit context referencing [27] represent complementary strategies that contribute to the integration of memory within semantic communication systems. These approaches enable semantic continuity and context preservation across transmissions, which are essential properties in sequential or multi-turn scenarios. The incorporation of memory mechanisms into the DeepSC architecture offers promising opportunities to enhance robustness and adaptability, although this remains an active area of research with several open challenges.

## 2.5 Transformer Encoders: BERT and DistilBERT

Transformer encoder architectures are widely used in semantic communication systems to generate dense, contextual embeddings from natural language input. Among the most influential models in this category is BERT (Bidirectional Encoder Representations from Transformers) [12]. BERT is pre-trained on large unlabeled corpora using two unsupervised objectives: masked language modeling (MLM) and next sentence prediction (NSP). The MLM task involves randomly masking a subset of tokens and training the model to predict the missing elements using the bidirectional context. The NSP task trains the model to determine whether a given sentence follows another in the original document, improving its ability to capture inter-sentence dependencies.

The input is tokenized using subword units and represented by a combination of token embeddings, segment embeddings, and positional encodings. The architecture of BERT-BASE consists of 12 transformer encoder layers, each containing multi-head self-attention and feed-forward sublayers. The model has 110 million parameters and provides strong performance across multiple language understanding tasks.

Due to its depth and size, BERT may not be suitable for applications with limited computational resources or strict latency constraints. To address this issue, DistilBERT was

introduced as a lighter alternative [36]. It is obtained through knowledge distillation, where a smaller student model learns to replicate the behavior of a larger teacher model. DistilBERT retains the same hidden size and attention configuration as BERT, but reduces the number of layers by 50% (from 12 to 6), resulting in 66 million parameters and a 60% improvement in inference speed.

The training objective for DistilBERT includes:

- a masked language modeling loss,
- a distillation loss that minimizes the divergence between the student and teacher output distributions,
- and a cosine embedding loss to align internal hidden representations.

These components ensure that the model preserves both functional behavior and representational structure, enabling the use of transformer encoders in settings where computational efficiency is essential.

Recent works have continued to refine the transformer encoder design. Warner et al. [3] propose a modernized bidirectional transformer architecture that improves generalization, training stability, and throughput. Their model incorporates optimizations such as improved normalization layers and enhanced scaling strategies. Chen et al. [13] explore a combined encoder-transformer approach that integrates encoder-derived representations with controlled generative decoding, enhancing coherence and semantic consistency in downstream tasks.

Transformer encoders such as BERT and DistilBERT are particularly effective when used as the semantic encoder module in neural communication systems. These models provide the ability to map discrete text inputs into latent semantic spaces that can be further processed and transmitted by the communication channel. The selection between full-scale models like BERT and lightweight variants such as DistilBERT depends on the balance between semantic precision and computational efficiency required by the target application.

*Table 1: Comparison of BERT and DistilBERT architectures*

| **Model** | **#Layers** | **Parameters** | **Relative Inference Time** |
|---|---|---|---|
| BERT-base [12] | 12 | 110M | 100% |
| DistilBERT [36] | 6 | 66M | 60% |

## 2.6 Semantic Evaluation Metrics

### 2.6.1 Reference-Based Metrics

In the evaluation of natural language generation systems such as machine translation, summarization, and semantic communication, reference-based metrics play a central role. These metrics estimate the quality of a candidate output by comparing it to one or more human-generated reference sentences. The core assumption is that a higher similarity to human references corresponds to greater linguistic quality and semantic fidelity.

Over the years, a variety of reference-based metrics have been introduced, each adopting a different approach to measure similarity. Earlier metrics rely on lexical overlap and surface-level patterns, while more recent ones incorporate semantic embeddings or model-based scoring functions. This section presents four prominent reference-based metrics that are frequently used in the literature and are particularly relevant for evaluating semantic quality: BLEU, METEOR, BERTScore, and BLEURT.

For each metric, we describe its main formulation, typical scoring scale, common applications, and known limitations. This overview offers a comprehensive foundation for understanding how automated evaluations are performed and what specific challenges arise when these metrics are applied to tasks where the preservation of meaning is a primary concern.

**BLEU.** The Bilingual Evaluation Understudy (BLEU) [20] is a reference-based metric originally developed for evaluating machine translation. It estimates the quality of a candidate sentence by comparing its $n$-gram overlap with one or more human-generated reference sentences. BLEU assumes that the closer a candidate is to a professional human translation, the better its quality.

BLEU combines modified $n$-gram precision with a brevity penalty (BP) to penalize excessively short translations. The score is computed as:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right),$$

where $p_n$ is the modified precision for $n$-grams, and $w_n$ are typically uniform weights. The

brevity penalty is defined as:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r, \\ e^{(1-r/c)} & \text{if } c \leq r, \end{cases}$$

with $c$ and $r$ denoting the lengths of the candidate and reference sentences, respectively.

BLEU scores range from 0 to 1 and are often reported as percentages. In practice, BLEU is computed over a corpus to improve reliability, as sentence-level scores can be unstable. On platforms like Hugging Face, implementations follow the WMT mteval-v13a tokenization by default, though other tokenizers can be used.

Despite its widespread use, BLEU presents several well-known limitations. It relies solely on precision and does not directly measure recall; the brevity penalty only partially compensates for this, and its adequacy remains debated. Furthermore, BLEU performs $n$-gram matching across all reference translations simultaneously, rather than comparing to each individually and selecting the best match, which may dilute alignment accuracy.

The metric also demands exact word matching, without accounting for stemming, morphological variants, or synonyms. This rigid criterion penalizes outputs that are semantically correct but lexically divergent. All $n$-gram matches are weighted equally, regardless of their linguistic importance, which can undervalue key content words. BLEU's use of geometric averaging across $n$-grams makes it highly sensitive to zero matches—if even one $n$-gram order has zero overlap, the entire score drops to zero.

Additionally, BLEU does not assess grammaticality or fluency, nor can it distinguish between critical and benign errors. For example, it may assign similar scores to sentences that preserve or invert meaning if $n$-gram overlap remains unchanged [28]. Because of these limitations, BLEU is often complemented with semantically-aware metrics such as BERTScore or BLEURT, especially in tasks where meaning preservation is crucial.

To assist interpretation, Table 2 summarizes approximate qualitative thresholds for BLEU scores, as suggested by the Google Cloud AutoML Translation documentation [6]. These categories help contextualize BLEU values in terms of perceived translation quality. For instance, scores below 10% typically indicate output that is nearly unintelligible, while values above 50% are generally associated with fluent and accurate translations. Although BLEU is

not designed to capture meaning explicitly, these intervals provide a practical reference when evaluating the relative performance of different systems on the same task and dataset.

*Table 2: Approximate interpretation of BLEU score ranges [6]*

| **BLEU (%)** | **Interpretation** |
|---|---|
| $< 10$ | Almost useless translation |
| 10–19 | Meaning is hard to understand |
| 20–29 | Message is clear but with major grammatical errors |
| 30–40 | Understandable to good-quality translations |
| 40–50 | High-quality translations |
| 50–60 | Very fluent and adequate translations |
| $> 60$ | Sometimes better than human translations |

**METEOR.** METEOR (Metric for Evaluation of Translation with Explicit ORdering) [2, 21] was introduced to address several limitations of BLEU, particularly its lack of recall and its insensitivity to word meaning and ordering. Unlike BLEU, which focuses on $n$-gram precision, METEOR computes an alignment between the candidate and reference translations based on flexible unigram-level matches, including exact matches, stemmed forms, and synonyms (via WordNet).

Once aligned, the metric calculates unigram-level *precision* (proportion of matched unigrams in the candidate) and *recall* (proportion of matched unigrams in the reference). These are combined using a weighted harmonic mean, known as F-mean:

$$F_{\text{mean}} = \frac{10 \cdot P \cdot R}{R + 9P},$$

which places more weight on recall, in line with findings that recall correlates better with human judgments [21].

To capture word order and fluency, METEOR applies a penalty based on *chunk fragmentation*. The fewer and longer the contiguous matched sequences (chunks), the lower the penalty:

$$\text{Penalty} = 0.5 \cdot \left( \frac{\#\text{chunks}}{\#\text{matched unigrams}} \right).$$

The final METEOR score is then:

$$\text{Score} = (1 - \text{Penalty}) \cdot F_{\text{mean}}.$$

Scores range from 0 to 1, with values closer to 1 indicating higher similarity to the reference. In practice, METEOR scores are typically higher than BLEU for the same system output, due to its more forgiving alignment and use of synonymy.

METEOR has been shown to correlate more strongly with human judgments than BLEU, especially at the sentence level. For example, in experiments on Arabic-English and Chinese-English translations from the DARPA/TIDES 2003 dataset, METEOR achieved system-level Pearson correlation coefficients as high as 0.964 [2]. The Pearson correlation coefficient (also known as Pearson's $r$) measures the linear relationship between two variables, in this case between automatic scores and human judgments; values close to 1 indicate strong agreement. Sentence-level correlations were lower, averaging around 0.331 and 0.347 for Chinese and Arabic, respectively, but still superior to those obtained with BLEU and NIST.

However, METEOR has known limitations. It can overestimate translation quality in the presence of superficial lexical similarity, and the weighting scheme for function words or synonyms can mask critical errors. Saadany and Orăsan [28] report that METEOR assigns high scores to mistranslations that omit sentiment-bearing words, such as negations, because of its tolerance toward matching via stemming or synonyms.

Despite these drawbacks, METEOR remains a strong baseline metric for tasks where partial semantic similarity and syntactic variation are common, especially in cases where more precise learned metrics like BLEURT are not available or too computationally expensive.

To help interpret the scores, Table 3 provides qualitative thresholds for METEOR values, as reported by Number Analytics [29]. These ranges offer a practical guideline for evaluating translation quality across systems and tasks.

*Table 3: Qualitative interpretation of METEOR score ranges [29]*

| **METEOR** | **Interpretation** |
|---|---|
| 0.0–0.2 | Very poor translation quality |
| 0.2–0.4 | Poor translation quality |
| 0.4–0.6 | Fair translation quality |
| 0.6–0.8 | Good translation quality |
| 0.8–1.0 | Excellent translation quality |

**BERTScore.** BERTScore [34] is a reference-based evaluation metric that leverages contextual embeddings from large pre-trained language models, such as BERT [12], to assess

semantic similarity between a candidate sentence and a reference. Unlike traditional metrics that rely on surface-level $n$-gram overlap, BERTScore compares token representations in vector space, enabling it to capture paraphrasing, synonymy, and flexible word order.

The metric computes token-level similarity using cosine distance between contextual embeddings. Precision is defined as the average maximum similarity between each candidate token and all reference tokens; recall is defined analogously:

$$\mathrm{P}_{\mathrm{BERT}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j} \max_{x_i} \cos(\hat{x}_j, x_i), \quad \mathrm{R}_{\mathrm{BERT}} = \frac{1}{|x|} \sum_{x_i} \max_{\hat{x}_j} \cos(x_i, \hat{x}_j)$$

The final BERTScore is their $F_1$-harmonic mean. To improve interpretability, scores are baseline-rescaled:

$$\hat{F}_1 = \frac{F_1 - b}{1 - b}$$

where $b$ is an empirically determined lower bound on unrelated sentence pairs.

BERTScore values range from 0 to 1, where higher values indicate stronger semantic alignment. Typical $F_1$ scores for well-aligned sentence pairs fall between 0.85 and 0.95, but scores can vary depending on model type, task, and language. In practice, BERTScore is often computed without baseline rescaling, and values near 1.0 represent high similarity, while lower values (below 0.7) indicate limited or noisy semantic overlap [34].

Despite its strengths, BERTScore inherits limitations from its underlying language model. As shown by Hanna and Bojar [9], BERTScore can struggle with linguistic phenomena such as negation, antonymy, and named entity disambiguation. In particular, it may assign overly high scores to incorrect candidates that are lexically or stylistically similar to the reference, even when they contain meaning-altering errors.

In semantic communication tasks, BERTScore offers a flexible and powerful tool to evaluate meaning preservation. However, due to its insensitivity to certain semantic divergences—especially when lexical overlap is high—it should be complemented with other metrics that better capture factual and functional correctness.

To assist with interpretation, Table 4 presents a qualitative guide to BERTScore values as commonly observed in NLP benchmarks.

*Table 4: Qualitative interpretation of BERTScore values*

| BERTScore | Interpretation |
|---|---|
| $< 0.70$ | Low semantic similarity |
| 0.70–0.85 | Moderate semantic similarity |
| 0.85–0.92 | Strong semantic similarity |
| $> 0.92$ | Very high or near-exact semantic alignment |

**BLEURT.** BLEURT (Bilingual Evaluation Understudy with Representations from Transformers) [32] is a learned reference-based metric designed to predict human judgments of text generation quality. It builds on the BERT architecture [12], which is fine-tuned to produce scalar quality scores for candidate-reference sentence pairs.

BLEURT operates in two phases: large-scale pre-training on synthetic perturbations of English sentences, followed by fine-tuning on human-annotated quality ratings. The pre-training step exposes the model to a broad range of plausible variations in grammar, style, and semantics by introducing artificial edits (e.g., synonym replacements, word deletions, backtranslation). These variations help the model learn generalizable representations of semantic equivalence and distortion.

Formally, given a reference sentence $x$ and a candidate $\tilde{x}$, BLEURT encodes both inputs using a shared BERT model and uses the [CLS] token embedding to predict a scalar score $\hat{y} \in \mathbb{R}$ via a regression layer:

$$\hat{y} = f(x, \tilde{x}) = \mathbf{W} \cdot \mathbf{v}_{[\text{CLS}]} + b$$

where $\mathbf{v}_{[\text{CLS}]}$ is the contextualized representation of the sentence pair and $\mathbf{W}$, $b$ are learnable parameters.

BLEURT's output has a range of values between -1 and approximately 1. This value indicates how similar the generated text is to the reference texts, with values closer to 1 representing more similar texts. In most practical scenarios, outputs range between 0.2 and 0.9. The model can be applied at the sentence or corpus level and also supports cases where multiple references are available: in such setups, BLEURT computes a score for each reference and returns the maximum value, assuming that the best-matching reference is the most informative.

Compared to lexical and embedding-based metrics, BLEURT has shown significantly higher correlation with human judgments across multiple years of the WMT Metrics Shared Task [32]. It outperforms BLEU, METEOR, and BERTScore in both Kendall's Tau [18] and Pearson correlation, especially for sentence-level evaluation.

Nonetheless, BLEURT has certain limitations. It is more computationally intensive than traditional metrics, requires access to trained models, and may generalize less effectively to unseen domains if the fine-tuning data is too narrow. However, thanks to its strong performance and robustness to quality drifts, it remains one of the most reliable metrics currently available for semantic evaluation of generated text.

To support practical interpretation, Table 5 offers an approximate guide to BLEURT scores based on common usage patterns in research and shared tasks.

*Table 5: Approximate interpretation of BLEURT score ranges*

| **BLEURT Score** | **Interpretation** |
|---|---|
| $< 0.3$ | Low semantic match; major divergence from reference |
| 0.3–0.5 | Acceptable quality; partial meaning overlap |
| 0.5–0.7 | Good match; most key information is preserved |
| 0.7–0.9 | High-quality output; minor differences from reference |
| $> 0.9$ | Near-perfect or better-than-reference match |

**Limitations of Reference-Based Metrics.** While BLEU, METEOR, and BERTScore have played an important role in the automatic evaluation of text generation systems, they suffer from critical limitations that reduce their reliability, especially in cases where semantic fidelity is more important than surface similarity.

Several studies, including the analysis by Saadany and Orăsan [28], have shown that these metrics often fail to penalize translations that introduce meaning-altering errors, especially in sentiment-bearing content. For example, translations that omit negation (e.g., rendering "May God not forgive you" as "May God forgive you") can receive high scores under all three metrics, despite conveying the opposite intent. BLEU and METEOR, which rely on $n$-gram overlap and soft lexical matches, assign scores above 0.70 in such cases due to preserved token similarity. Even BERTScore, despite its semantic basis, can assign inflated scores because antonyms like "great" and "terrible" may appear close in embedding space.

Figure 22 illustrates this problem. It shows how the three metrics rate candidate trans-

lations with both non-critical and critical sentiment errors. Notably, the scores remain high and relatively uniform, even when the semantic distortion is substantial.

| Synthetic Data | | Metric | | |
|---|---|---|---|---|
| | | BLEU | METEOR | BERTScore |
| Ref | Their pizza is the best, if you like thin crusted pizza. | 1.0 | 1.0 | 1.0 |
| Non-critical Error | Their pizza is the best, if you like thin ***layer*** pizza. | 0.76 | 0.50 | 0.90 |
| Critical Error | Their pizza is the ***worst***, if you like thin crusted pizza. | 0.73 | 0.50 | 0.86 |
| Authentic Data | | | | |
| Ref | What is this amount of happiness, I don't understand! | 1.0 | 1.0 | 1.0 |
| One Error | What is this amount of ***anger***, I don't get it! | 0.65 | 0.47 | 0.89 |
| Ref | Sweetie like clouds, always fill me with joy. | 1.0 | 1.0 | 1.0 |
| No Error | ***My love*** is like clouds, always fill me with joy. | 0.65 | 0.44 | 0.52 |

*Figure 22: Segment-level scores for candidate translations with non-critical and critical sentiment errors, evaluated by BLEU, METEOR, and BERTScore. Adapted from [28].*

As shown in the figure, all three metrics assign relatively high scores even to translations containing meaning-reversing errors such as dropped negations. This indicates that the metrics are not sufficiently sensitive to semantic fidelity and tend to over-rely on lexical or embedding similarity. The inability to penalize critical shifts in meaning raises concerns about their reliability in contexts where preserving intent is essential.

These shortcomings are especially concerning in tasks like semantic communication, where accurate intent transmission matters more than lexical fidelity. Traditional metrics tend to overvalue surface similarity and fail to distinguish between meaning-preserving and meaning-breaking variations.

To overcome these issues, learned metrics such as BLEURT have been introduced. Trained directly on human-annotated quality scores and pre-trained on synthetic data to handle a wide range of distortions, BLEURT demonstrates greater sensitivity to subtle semantic errors. As shown in shared evaluation benchmarks like WMT, BLEURT consistently achieves stronger correlation with human judgment at both system and sentence level. This makes it a more robust and context-aware tool for evaluating semantic quality in language generation.

Table 6 summarizes the main characteristics, score ranges, and known limitations of the reference-based metrics discussed in this section. This comparative overview serves both as a quick reference and as a basis for motivating the need for more semantically aware evaluation approaches.

*Table 6: Summary of reference-based evaluation metrics.*

| Metric | Basis | Score Range | Main Limitation |
|---|---|---|---|
| BLEU [20] | $n$-gram overlap | 0–1 | Ignores meaning |
| METEOR [2] | Unigram + synonyms | 0–1 | Overvalues surface match |
| BERTScore [34] | BERT embeddings | 0–1 | Weak on negation, antonyms |
| BLEURT [32] | Fine-tuned BERT | -1∼1 | Requires large models and tuning |

### 2.6.2 Reference-Free Metrics

As highlighted in recent literature [31], reference-free metrics represent a growing field in the evaluation of natural language generation (NLG) systems. These approaches aim to estimate the quality of generated texts without relying on human-written reference sentences. This direction is especially valuable in open-ended or low-resource settings, where collecting gold-standard references is costly or impractical.

In the context of semantic communication, reference-free evaluation provides a powerful tool for assessing fluency, coherence, and grammaticality at runtime—without requiring an external reference. These metrics are typically classified into two broad categories [31]: (i) hypothesis-only evaluation, which examines textual quality based solely on the generated output, and (ii) context-aware evaluation, which considers the alignment between the input and the output. In this case, even though no explicit reference is available, the input provides a form of background knowledge or semantic frame that guides the interpretation and evaluation of the output. In this section, we focus on two representative metrics from the former category: perplexity-based scores and grammatical acceptability classifiers.

**Perplexity (PPL).** Perplexity is one of the most established metrics for evaluating the fluency and predictability of language models. It measures the inverse likelihood of a

generated word sequence, defined as:

$$\text{PPL} = \left( \prod_{t=1}^{T} P(w_t | w_{1:t-1}) \right)^{-\frac{1}{T}},$$

where $w_1, \ldots, w_T$ is the output sequence. Lower perplexity indicates better alignment with the training distribution.

Recent work [39] demonstrates that perplexity can also serve as a useful tool for data pruning: small models can score training samples by perplexity and effectively improve the performance of much larger models. However, their results also highlight a limitation—models trained on data with lower test perplexity do not always perform better on downstream benchmarks, raising concerns about perplexity's reliability as a standalone quality metric.

The range of this metric is $[0, \infty)$, with lower values indicating better predictive performance. However, as highlighted by [24], perplexity values are highly dependent on the specific model and training corpus. This means that perplexity scores are not directly comparable across different models or datasets, limiting their utility in cross-system evaluation.

This insight motivates the development of more robust alternatives like unigram-normalized perplexity.

**Unigram-Normalized Perplexity (PPLu).** To address the limitations of standard perplexity, Roha et al. [17] introduced *Unigram-Normalized Perplexity (PPLu)*, a metric designed to be invariant to vocabulary size and more reflective of contextual modeling. It is defined as:

$$\text{PPLu} = \left( \prod_{t=1}^{T} \frac{P(w_t | w_{1:t-1})}{P(w_t)} \right)^{-\frac{1}{T}}.$$

By normalizing each conditional probability by the unigram frequency of the predicted word, PPLu quantifies the information gain from context over a unigram baseline. The logarithmic form of PPLu corresponds to mutual information:

$$\log \text{PPLu} = -\frac{1}{T} \sum_{t=1}^{T} \log \frac{P(w_t, w_{1:t-1})}{P(w_t) P(w_{1:t-1})}.$$

Empirical results show that PPLu is more stable across datasets with different vocabularies and more reliable in ranking sentence quality, especially when frequent tokens skew raw perplexity [17]. Unlike standard perplexity, it better distinguishes between true contextual understanding and statistical frequency matching.

**Acceptability Classification via CoLA.** Another class of reference-free metrics assesses whether a generated sentence is grammatically acceptable in the target language. A canonical resource for this task is the **Corpus of Linguistic Acceptability (CoLA)** [1], which contains over 10,000 English sentences annotated as acceptable or unacceptable based on linguistic judgments from the theoretical literature.

Acceptability classifiers trained on CoLA are typically binary sentence classifiers, often implemented using deep language models such as BERT. These models are fine-tuned to output a scalar score indicating the likelihood that a sentence conforms to the syntactic and semantic norms of English. Accuracy is typically evaluated using metrics such as Matthews Correlation Coefficient (MCC), where human-level agreement reaches around 0.70 [1].

In semantic communication, such models serve as valuable proxies for syntactic well-formedness and plausibility of reconstructed messages. Unlike perplexity-based metrics, CoLA-based classifiers are sensitive to phenomena such as argument structure, question formation, and binding, making them suitable for verifying that generated outputs not only are fluent but also grammatically sound.

Table 7 provides a compact comparison of the main reference-free metrics discussed in this section, highlighting their scope, scale, and interpretability.

*Table 7: Summary of reference-free evaluation metrics.*

| **Metric** | **Basis** | **Score Range** | **Limitation** |
|---|---|---|---|
| PPL [39] | Language model prediction accuracy | $(0, \infty)$; lower is better | Vocabulary-dependent; not comparable across setups |
| PPLu [17] | Contextual informativeness (MI-normalized) | $(0, \infty)$; lower is better | Requires external unigram estimates |
| CoLA [1] | Grammatical acceptability classification | Binary classifier; MCC $\sim 0.70$ | Requires task-specific training; sensitive to syntax phenomena |

# 3 Methodology

## 3.1 System Overview

The architecture adopted in this work extends the original Transformer-based DeepSC framework described in Section 2.2, which models semantic communication as a differentiable end-to-end learning problem. While the baseline system demonstrates the feasibility of transmitting natural language over noisy channels, the current implementation introduces several enhancements aimed at increasing robustness, adaptability, and realism in practical scenarios.

The core encoder–channel–decoder pipeline is preserved, but it is augmented with additional components that improve both the inference process and the evaluation loop. Specifically, the system integrates:

- **Improved Reference-Based Metrics:** BLEURT and BERTScore are used during testing to assess the semantic fidelity between transmitted and received messages. These replace the previous cosine similarity approach and provide more sensitive, fine-grained evaluations.

- **Reference-Free Semantic Decision:** During simulation, the receiver evaluates the quality of decoded sentences using internal, reference-free metrics (PPL, PPLu, and CoLA). Sentences falling below predefined thresholds are marked as semantically unacceptable and may trigger retransmission.

- **Retransmission Loop (Two Variants):** Two feedback strategies are implemented. The first is a *baseline retransmission loop*, where the same model handles up to three independent decoding attempts using separately transmitted signals. No memory is retained between transmissions. The second is a *memory-augmented loop*, in which a dedicated retransmission model (TX2) performs a single additional decoding step that fuses the received signal from the initial transmission (TX1) with its own, allowing the system to refine the output based on prior communication history.

These components are designed to approximate realistic feedback mechanisms in practical

semantic communication systems, especially in low-SNR or resource-constrained settings. The following subsections provide a detailed description of each module.

## 3.2 Improved Reference-Based Metrics

The original DeepSC validation relied on a custom cosine similarity between the average word embeddings of predicted and reference sentences. This approach, while computationally efficient, had several limitations:

- It ignored word order, treating permutations of the same words as equivalent.
- It lost contextual nuance by collapsing all tokens into a single average vector.
- It was not sensitive to critical semantic errors such as negations or word substitutions.

To overcome these issues, we integrated more robust semantic metrics:

- **BLEURT**, a learned evaluation model that correlates strongly with human judgments.
- **BERTScore**, which measures semantic similarity based on contextual token embeddings.
- **BLEU**, maintained for backward compatibility and comparability with prior benchmarks.

These metrics are used not only for validation but also in the final evaluation pipeline, providing a more accurate measure of semantic fidelity across different transmission conditions.

## 3.3 Reference-Free Semantic Decision

In real-world scenarios, semantic communication systems must often operate without access to ground-truth reference sentences. To enable automatic retransmission decisions in such settings, this work adopts a reference-free evaluation strategy based on three metrics: **Perplexity (PPL)**, **Unigram-Normalized Perplexity (PPLu)**, and **grammatical acceptability via CoLA**. These metrics were further detailed in Section 2.6.2, which provides formal definitions and a comparison of their properties.

The goal is to determine whether a decoded sentence should be accepted or discarded and retransmitted, based solely on its intrinsic quality. To this end, a decision logic is implemented that evaluates each decoded output along three dimensions:

- **Perplexity (PPL)** is computed using the GPT-2 language model, following standard formulations where lower values indicate greater fluency and compatibility with natural language. The implementation uses the Hugging Face model `gpt2`.
- **Unigram-Normalized Perplexity (PPLu)** provides a context-aware measure of semantic coherence by comparing the conditional probability of the sentence to its unigram likelihood. This accounts for the informativeness gained from context and is robust to vocabulary frequency effects.
- **Grammatical acceptability** is assessed using a binary classifier fine-tuned on the CoLA dataset, implemented via the `textattack/roberta-base-CoLA` model. The output predicts whether the sentence is linguistically well-formed.

The system uses a threshold-based logic to make decisions that prioritize semantic fidelity over strict grammaticality. Specifically, the decision policy is as follows:

1. If PPL is below a conservative threshold (PPL $<$ 150), the sentence is accepted unconditionally, as it is considered sufficiently fluent.
2. If PPL is moderate (150 $\leq$ PPL $<$ 500), the sentence is accepted only if it also passes the CoLA acceptability test.
3. If PPL is high ($\geq$ 500), the sentence may still be accepted if it exhibits high semantic informativeness (PPLu $<$ 10) *and* passes CoLA.

This multi-tiered strategy ensures that sentences are not rejected solely for minor grammatical errors, provided they carry sufficient contextual and semantic meaning. The thresholds were empirically selected based on preliminary analysis of typical values in GPT-2 outputs and human-labeled acceptability predictions from CoLA. While the PPL bounds aim to capture general language fluency, the PPLu condition offers a safeguard for semantically meaningful but structurally atypical sequences.

Overall, this reference-free feedback mechanism enables the decoder to autonomously assess the quality of its outputs in the absence of references, making the system more adaptable to real-time and resource-constrained deployment environments.

## 3.4 Retransmission Loop

To better model retransmission behavior in practical semantic communication systems, this work implements and compares two variants of the retransmission loop:

- A **baseline strategy**, where the same model (TX1) is used for both the initial transmission and any retransmissions. Each decoding attempt operates independently, without leveraging information from prior transmissions.
- An **enhanced memory-aware strategy**, where a second transmission model (TX2) is introduced. In this case, TX2 receives both the new signal and the previously received one from TX1, enabling a more informed decoding based on concatenated features.

The two variants are described below.

### 3.4.1 Baseline Strategy

In the baseline retransmission scheme, the same encoder–decoder model (TX1) is reused across all transmission attempts. Each decoding is performed independently: the source sentence is re-encoded and retransmitted from scratch. No memory of prior attempts is retained, and no fusion between received signals is performed.

**Training and Validation.** Training in the baseline strategy follows the standard DeepSC pipeline. For each sentence in the batch, the model encodes the source input, applies channel encoding, and transmits the signal through a noisy channel. The received signal is decoded to produce a predicted output, which is compared against the ground-truth to compute the reconstruction loss. If mutual information regularization is used, the corresponding loss term is added, and the Mutual information (MI) network is used in evaluation mode, as detailed in Algorithm 1.

Validation reuses the same forward architecture, but no gradient updates are performed. All model components operate in inference mode, and mutual information (if enabled) is

included only as a diagnostic term. The validation loss is used to monitor model performance and guide early stopping, as described in Algorithm 2.

**Algorithm 1** Training loop for baseline model (TX1 only)

**for** each sentence $s$ in the training batch **do**
    Encode $s$ using TX1 encoder
    Transmit over noisy channel → obtain RX
    Decode RX and compute output $\hat{s}$ using TX1 decoder
    Compute cross-entropy loss between $\hat{s}$ and reference $s$
    **if** `mi_net` is provided **then**
        Compute mutual information loss and add it to the total loss
    **end if**
    Backpropagate and update TX1 parameters
**end for**

**Algorithm 2** Validation loop for baseline model (no updates)

**for** each sentence $s$ in the validation batch **do**
    Encode $s$ using TX1 encoder
    Transmit over noisy channel → obtain RX
    Decode RX and compute output $\hat{s}$ using TX1 decoder
    Compute validation loss between $\hat{s}$ and reference $s$
    **if** `mi_net` is provided **then**
        Add mutual information loss term (no gradient update)
    **end if**
**end for**

**Testing Phase.** At test time, the system uses TX1 for all decoding attempts. After each transmission, the decoded output is evaluated using the reference-free semantic acceptability logic described in Section 3.3. If the sentence fails, a new transmission is issued, up to a maximum of three attempts. The first acceptable output is retained; otherwise, the final attempt is accepted by default. The semantic ARQ logic for baseline retransmission is summarized in Algorithm 3.

### 3.4.2 Memory-Aware Strategy

In the enhanced retransmission strategy, the system introduces a second encoder–decoder model (TX2) dedicated to handling retransmissions. Unlike the baseline case, the second decoding attempt is informed by both the current and previous received signals, allowing the system to refine its output based on communication history.

**Algorithm 3** Baseline ARQ Retransmission Logic

**Input:** Source sentence $s$
**for** attempt $i = 1$ to 3 **do**
    Encode and transmit $s$ using TX1
    Decode output $\hat{s}_i$
    **if** $\hat{s}_i$ passes acceptability test **then**
        Accept $\hat{s}_i$ and exit
    **end if**
**end for**
Accept $\hat{s}_3$ (max attempts reached)

This memory-aware scheme enables the receiver to fuse information across attempts, improving reconstruction quality in challenging channel conditions.

**Training and Validation.** The training pipeline involves two sequential transmissions of the same source sentence. TX1 performs the first transmission and remains frozen. TX2 performs a second transmission and fuses its received signal with RX1 (from TX1) to enhance semantic recovery. Only TX2 is updated via backpropagation during training, as detailed in Algorithm 4.

Validation follows the same processing flow, but with no weight updates. If a mutual information module is used, its contribution is included during both training and validation, but the MI network remains frozen during validation, as described in Algorithm 5.

**Algorithm 4** Training loop for memory-aware model (TX1 frozen, TX2 updated)

**for** each sentence $s$ in the training batch **do**
    Encode $s$ using TX1 encoder (frozen) $\rightarrow$ obtain RX1
    Encode $s$ using TX2 encoder $\rightarrow$ obtain RX2
    Concatenate RX1 and RX2: $\text{RX}_{\text{concat}} = [\text{RX1}; \text{RX2}]$
    Decode $\text{RX}_{\text{concat}}$ and compute output $\hat{s}$ using TX2
    Compute cross-entropy loss between $\hat{s}$ and reference $s$
    **if** `mi_net` is provided **then**
        Compute mutual information loss and add it to the total loss
    **end if**
    Backpropagate and update TX2 parameters
**end for**

**Algorithm 5** Validation loop for memory-aware model (no updates)

**for** each sentence $s$ in the validation batch **do**
    Encode $s$ using TX1 encoder (frozen) $\rightarrow$ obtain RX1
    Encode $s$ using TX2 encoder (frozen) $\rightarrow$ obtain RX2
    Concatenate RX1 and RX2: $\text{RX}_{\text{concat}} = [\text{RX1}; \text{RX2}]$
    Decode $\text{RX}_{\text{concat}}$ and compute output $\hat{s}$ using TX2
    Compute validation loss between $\hat{s}$ and reference $s$
    **if** `mi_net` is provided **then**
        Add mutual information loss term (no gradient update)
    **end if**
**end for**

**Testing Phase.** During testing, retransmission is handled dynamically:

1. A sentence is transmitted via TX1 and decoded.
2. If the output fails the reference-free acceptability test, a retransmission is issued.
3. The retransmission is sent via TX2; the two received signals (from TX1 and TX2) are concatenated and decoded.
4. If the second output still fails, it is accepted by default. The system allows a **maximum of two transmission attempts per sentence**.

This process emulates a realistic ARQ feedback loop without ground-truth, enabling sentence-wise adaptive retransmission based solely on intrinsic quality signals.

The logic is summarized in Algorithm 6.

**Algorithm 6** Semantic ARQ Retransmission Logic

**Input:** Source sentence $s$
Encode and transmit $s$ with TX1 $\rightarrow$ obtain RX1
Decode output $\hat{s}_1$
**if** $\hat{s}_1$ fails semantic acceptability test **then**
    Encode and transmit $s$ with TX2 $\rightarrow$ obtain RX2
    Concatenate RX1 and RX2: $\mathrm{RX}_{\mathrm{concat}} = [\mathrm{RX1}; \mathrm{RX2}]$
    Decode $\mathrm{RX}_{\mathrm{concat}}$ and compute output $\hat{s}$ using TX2
    **if** $\hat{s}_2$ fails acceptability test **then**
        Accept $\hat{s}_2$ (max attempts reached)
    **else**
        Accept $\hat{s}_2$
    **end if**
**else**
    Accept $\hat{s}_1$
**end if**

## 3.5 Memory-Augmented Decoding

In conventional retransmission schemes, each decoding attempt is treated independently, with no memory of past transmission efforts. In contrast, the memory-augmented decoding strategy adopted in this work introduces a lightweight form of memory by fusing received signals across transmission attempts.

Specifically, the decoder of the retransmission model (TX2) receives as input the concatenation of the received signal from the initial transmission (TX1) and the one from its own transmission. This results in a composite tensor $\mathrm{RX}_{\mathrm{concat}} = [\mathrm{RX1}; \mathrm{RX2}] \in \mathbb{R}^{B \times L \times 2d}$, where $B$ is the batch size, $L$ the sequence length, and $d$ the feature dimension.

By conditioning its prediction on both signals, TX2 is implicitly able to incorporate information from past attempts. This concatenated input acts as a memory buffer, allowing the decoder to leverage contextual cues, partial signal content, or redundancy from the previous pass. Unlike architectures that require explicit memory states (e.g., recurrent networks or attention over prior outputs), this approach offers a memory mechanism entirely embedded in the input representation.

This design improves robustness in noisy channels and enables the model to correct errors from previous transmissions. The decoder effectively learns to interpret the joint signal space as a richer semantic prior, supporting refined and more accurate reconstructions.

## 3.6 Dataset: Europarl Corpus

All experiments in this work are conducted on the **Europarl** corpus, a large-scale parallel dataset composed of proceedings from the European Parliament. This dataset is well-suited for evaluating semantic communication in natural language settings due to its formal tone, sentence-level segmentation, and topical variety.

We use a preprocessed version of the English monolingual subset, which is tokenized and stored in `.pkl` format for efficient loading. The dataset is split into training, validation, and test sets. Each data sample consists of a tokenized sentence represented as a list of word indices, compatible with the vocabulary of the Transformer encoder-decoder model.

The dataset class `EurDataset` handles loading and indexing. During training, batches are padded to the length of the longest sentence and sorted by length to improve efficiency during Transformer-based decoding.

The Europarl corpus offers several advantages for semantic communication studies:

- It contains high-quality, well-formed natural language sentences;
- It supports both sentence-level and discourse-level semantic evaluation;
- It enables realistic testing of retransmission strategies based on grammar and semantic coherence.

The use of Europarl ensures that the system is tested on a linguistically rich and domain-relevant dataset, offering insights into the real-world applicability of the proposed feedback and memory-enhanced DeepSC system.

## 3.7 Post-processing and Evaluation Pipeline

To support the analysis of model behavior under different channel conditions and retransmission strategies, this work includes a custom post-processing pipeline that parses simulation outputs and generates comparative performance plots.

The pipeline processes two main sources of output:

- `simresults.log`: a log file containing training and validation losses (including cross-entropy and mutual information) along with instantaneous SNR values per epoch.

- `performance_results.txt`: a summary file reporting, for each SINR value, the average semantic quality metrics and the average number of retransmissions (`avg_RTX`) under each configuration (NoRETX, baseline, memory-aware).

Based on this data, the pipeline performs the following analyses:

- **Metric comparison across SINR:** For each evaluation metric (BLEU, BLEURT, BERTScore, PPL, PPLu, CoLA), the script generates plots that compare performance across SINR values for three configurations: No retransmission, baseline retransmission, and memory-aware retransmission.
- **Percentage improvement computation:** For each SINR point and metric, the script computes the percentage gain (or loss) of both baseline and memory-aware strategies with respect to NoRETX. These results are saved in separate reports for reference-based and reference-free metrics.
- **Exporting visualizations and reports:** All plots and textual summaries are automatically saved in the specified output directory, with separate files for each metric and channel condition.

This post-processing pipeline enables reproducible and efficient evaluation of model behavior, highlighting how semantic quality varies as a function of SNR and retransmission strategy. By directly comparing NoRETX, baseline, and memory-aware configurations, it provides clear insight into the semantic gains introduced by feedback and memory mechanisms.

# 4 Results

This section presents the results obtained under two channel models: CDL-B and Rayleigh. For each scenario, the baseline and memory-aware configurations are evaluated both with and without retransmission. The analysis includes reference-based metrics (BLEU, BLEURT, BERTScore) and reference-free indicators (PPL, PPLU, and CoLA), offering a comprehensive view of semantic fidelity and linguistic quality across varying SINR conditions.

## 4.1 CDL-B Channel

### Reference-Based Metrics

Figures 23, 24, and 25 report BLEU, BLEURT, and BERTScore under CDL-B conditions, comparing baseline and memory-aware configurations, both with and without retransmission.

In the baseline system, retransmission introduces only marginal differences, without consistent improvements. As shown in Figure 23, BLEU scores remain nearly flat beyond 4dB, with negligible variation between retransmission and no-retransmission setups. BLEURT shows similarly minor fluctuations, but achieves slightly higher gains than BLEU, particularly at lower SINR values. BERTScore, on the other hand, shows slightly more consistent gains, although the improvements remain limited. As summarized in Table 8, BERTScore achieves an improvement of approximately 0.5% across SINR levels, outperforming BLEU and BLEURT, whose gains are generally below **0.13%** and **0.3%** respectively. The only exception is BLEURT at -4dB, which reaches a peak improvement of 0.641%. These results indicate that, in the absence of memory, retransmission alone does not significantly enhance semantic fidelity.

In contrast, the memory-aware configuration shows a higher benefit from retransmission. As reported in Table 8, BLEU improves by +1.149% at -4dB and maintains gains above 1% across all SINR levels. BLEURT shows stronger improvements, reaching +2.672% at -4dB and stabilizing around +2.6% at higher SINR values. BERTScore also benefits from memory, with gains between +1.939% and +2.151%. This improvement across all three metrics confirms that memory integration enables the system to accumulate and refine semantic information over repeated transmissions, making retransmission effective only when

combined with memory-awareness.

Furthermore, as expected, all configurations show a general upward trend in metric scores as SINR increases. Higher SINR levels correspond to better channel conditions, leading to more accurate decoding and, consequently, higher semantic fidelity regardless of whether retransmission or memory is used.

Table 8 quantifies these trends by reporting the percentage improvement of both baseline and memory-aware configurations over the NoRETX setup, across all SINR levels and reference-based metrics.

*Table 8: Percentage improvement over NoRETX for BLEU, BLEURT, and BERTScore under CDL-B conditions.*

| 2*SINR [dB] | BLEU | | BLEURT | | BERTScore | |
|---|---|---|---|---|---|---|
| | Baseline | Memory | Baseline | Memory | Baseline | Memory |
| -4 | 0.133% | 1.149% | 0.641% | 2.672% | 0.504% | 1.939% |
| 0 | 0.058% | 1.031% | 0.301% | 2.450% | 0.564% | 1.937% |
| 4 | 0.016% | 0.992% | 0.188% | 2.395% | 0.558% | 1.997% |
| 8 | 0.031% | 1.034% | 0.197% | 2.492% | 0.575% | 2.076% |
| 12 | 0.015% | 1.055% | 0.108% | 2.555% | 0.560% | 2.104% |
| 16 | 0.011% | 1.066% | 0.076% | 2.655% | 0.560% | 2.151% |
| 20 | 0.007% | 1.059% | 0.086% | 2.636% | 0.544% | 2.148% |

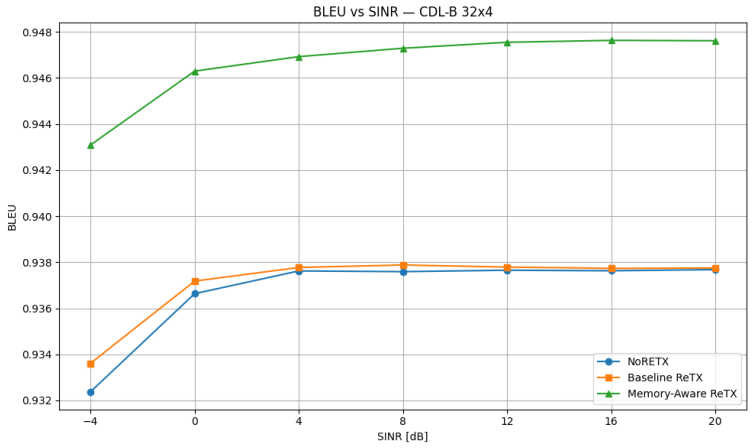



*Figure 23: BLEU scores under CDL-B conditions comparing NoRETX, FullRETX (baseline), and FullRETX (memory-aware).*

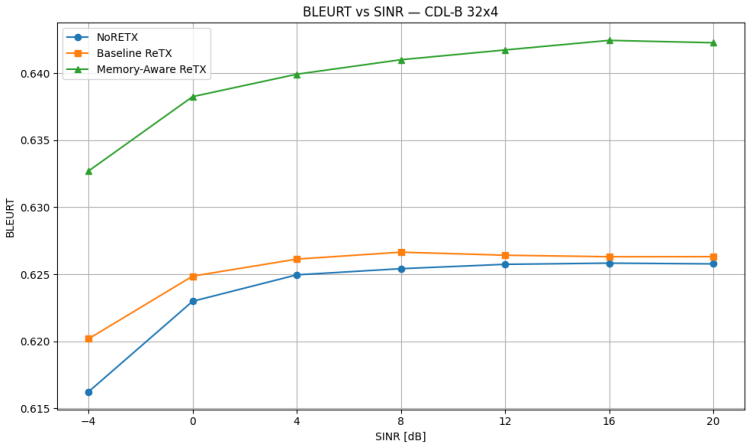



*Figure 24: BLEURT scores under CDL-B conditions comparing NoRETX, FullRETX (baseline), and FullRETX (memory-aware).*

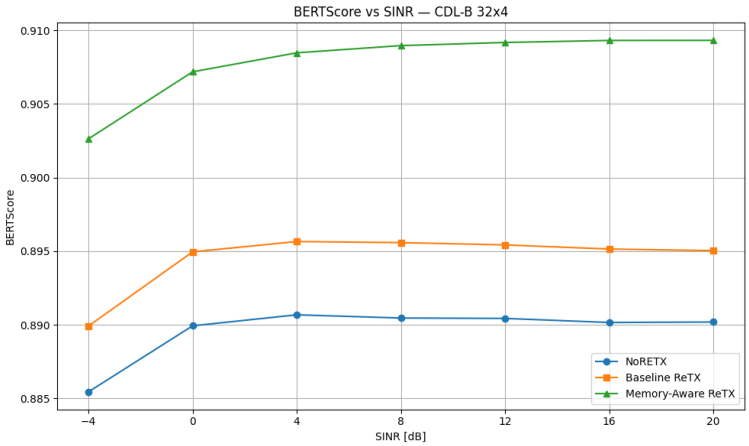



*Figure 25: BERTScore under CDL-B conditions comparing NoRETX, FullRETX (baseline), and FullRETX (memory-aware).*

### Reference-Free Metrics

Figures 26, 27, and 28 report Perplexity (PPL), unigram-level Perplexity (PPLU), and CoLA acceptability under CDL-B conditions, comparing baseline and memory-aware configurations, both with and without retransmission.

Unlike reference-based metrics, where higher values indicate better performance, in the case of PPL and PPLU, lower values are preferred, as they reflect higher confidence and fluency in the language model's predictions.

For PPL, the memory-aware configuration shows some improvements from retransmission. As summarized in Table 9, it achieves a peak reduction of **13.3%** at -4dB, and

maintains gains above **9.5%** across all SINR levels. These results suggest the benefit of memory mechanisms in refining sentence-level fluency during repeated decoding.

In contrast, the baseline system exhibits poor behavior under retransmission for PPL. At -4dB, no value is reported (N/A), likely due to numerical instability or undefined decoding. Furthermore, at 0dB, PPL performance actually worsens by **+0.606%** compared to NoRETX, and overall improvements remain marginal, with the best gain being only **2.13%** at 4dB. This confirms that without memory, retransmission struggles to stabilize sentence-level predictions and may even degrade them under moderate noise.

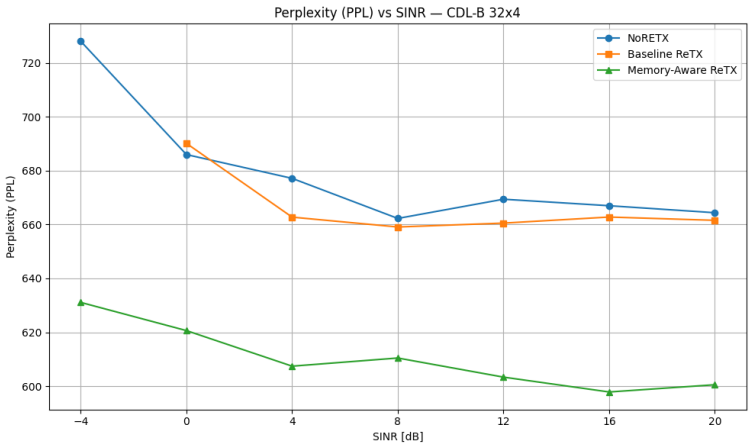



*Figure 26: PPL under CDL-B conditions comparing NoRETX, FullRETX (baseline), and FullRETX (memory-aware). Lower values are better.*

PPLU values, which reflect unigram-level token predictability, remain mostly stable across configurations. The baseline system without retransmission consistently achieves the lowest PPLU values, and the addition of retransmission introduces only minor improvements. As shown in Table 9, the improvements remain below **0.03%** in all cases. Specifically, the baseline configuration shows consistent reductions in PPLU across all SINR values (e.g., **-0.028%** at -4dB), indicating slightly improved token predictability.

The memory-aware configuration also exhibits small gains at low SINR values, but shows slight regressions at higher SINRs, with PPLU increasing by up to **+0.029%** at 20dB. These results suggest that while retransmission with memory improves sentence-level fluency (as seen with PPL), it does not consistently enhance local token-level predictability, and may even introduce minimal degradation under clean channel conditions, which is unexpected and might possibly depend on statistical limitations of the testing dataset.

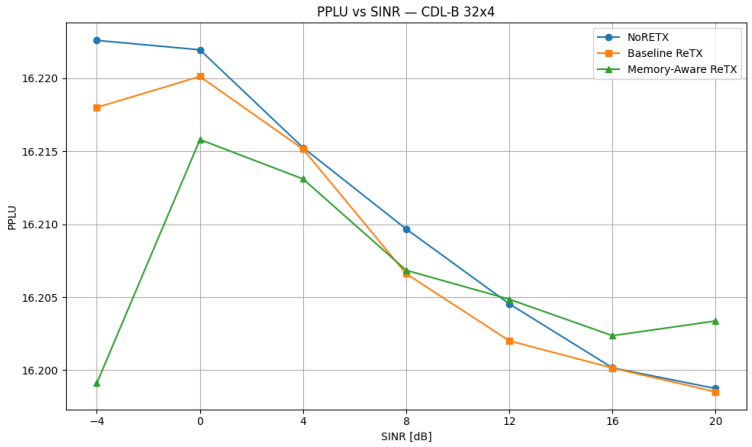



*Figure 27: PPLU under CDL-B conditions comparing NoRETX, FullRETX (baseline), and FullRETX (memory-aware). Lower values are better.*

CoLA acceptability, on the other hand, improves more significantly across both configurations. In the baseline system, retransmission yields a notable gain of **+10.3%** at -4dB, which remains slightly higher than the memory-aware improvement of **+8.7%** at the same SINR. A similar advantage holds at 0dB, where baseline scores slightly exceed those of the memory-aware setup. However, as SINR increases, the baseline configuration exhibits a counterintuitive trend: CoLA scores progressively decrease, reaching only **+1.3%** improvement at 20dB. This suggests that retransmission alone, without memory, fails to consistently benefit grammatical acceptability as channel conditions improve.

In contrast, both the NoRETX and memory-aware configurations follow the expected trend: CoLA acceptability increases steadily with SINR. The memory-aware system, in particular, maintains stable and strong improvements across all SINR values, ranging from **+6.7%** to **+8.7%**. These results indicate that retransmission is more effective at recovering grammatical structure when coupled with memory mechanisms, and that memory integration helps sustain syntactic coherence even as channel quality improves.

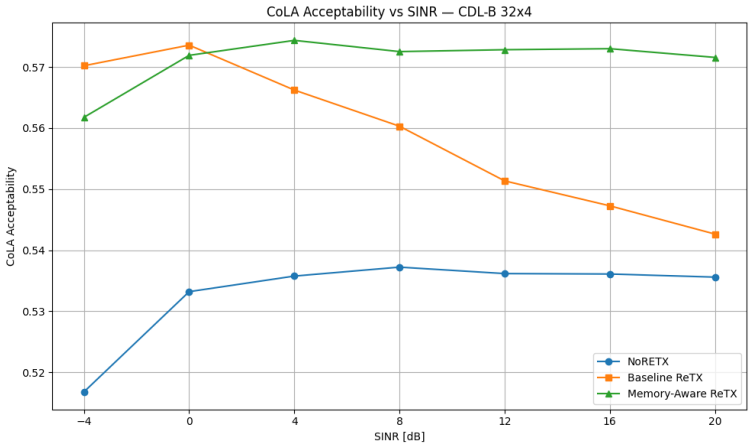



*Figure 28: CoLA acceptability under CDL-B conditions comparing NoRETX, FullRETX (baseline), and FullRETX (memory-aware). Higher values are better.*

Table 9 quantifies these trends by reporting the percentage improvement of both baseline and memory-aware configurations over the NoRETX setup, across all SINR levels and reference-free metrics.

*Table 9: Percentage improvement over NoRETX for PPL, PPLU, and CoLA under CDL-B conditions. For PPL and PPLU, lower values indicate better performance, so negative percentages imply improvement.*

| 2*SINR [dB] | PPL | | PPLU | | CoLA | |
|---|---|---|---|---|---|---|
| | Baseline | Memory | Baseline | Memory | Baseline | Memory |
| -4 | N/A | -13.315% | -0.028% | -0.145% | +10.331% | +8.698% |
| 0 | +0.606% | -9.525% | -0.011% | -0.038% | +7.569% | +7.256% |
| 4 | -2.126% | -10.296% | -0.001% | -0.013% | +5.684% | +7.202% |
| 8 | -0.479% | -7.824% | -0.019% | -0.017% | +4.294% | +6.568% |
| 12 | -1.325% | -9.865% | -0.016% | +0.002% | +2.830% | +6.835% |
| 16 | -0.630% | -10.366% | -0.000% | +0.014% | +2.082% | +6.880% |
| 20 | -0.425% | -9.611% | -0.001% | +0.029% | +1.315% | +6.715% |

## 4.2 Rayleigh Channel

### Reference-Based Metrics

Figures 29, 30, and 31 present BLEU, BLEURT, and BERTScore under Rayleigh fading conditions. Results are shown for both baseline and memory-aware configurations, comparing retransmission (RETX) and no retransmission (NoRETX).

In the baseline configuration, retransmission leads to consistent but modest improvements

across all metrics. As shown in Figure 29, BLEU scores increase slightly at low SINR values, with a peak gain of **0.534%** at -4dB, then converge toward the no-retransmission curve as SINR increases. BLEURT and BERTScore follow similar trends: BLEURT reaches a maximum improvement of **1.854%**, while BERTScore peaks at **1.159%**, both at -4dB. However, these benefits diminish steadily, with improvements dropping below **0.1%** at high SINR values. These results suggest that retransmission provides some marginal benefit in noisy conditions but becomes ineffective as channel quality improves.

The memory-aware configuration shows stronger and more stable gains. BLEU improves by **1.970%** at -4dB and maintains gains above **1.1%** across the entire SINR range. BLEURT benefits even more noticeably, with an improvement of **4.199%** at -4dB and sustained gains around **2.1–2.4%** at higher SINR values. Similarly, BERTScore rises by **3.147%** at low SINR and remains above **1.8%** even at 20dB. These trends confirm that memory mechanisms allow the system to effectively leverage repeated decoding attempts, improving semantic alignment consistently across channel conditions.

As expected, all configurations show a general upward trend in metric scores as SINR increases. Better channel conditions lead to more accurate decoding, thereby enhancing semantic fidelity regardless of retransmission or memory integration.

Table 10 quantifies these trends by reporting the percentage improvement of both baseline and memory-aware configurations over the NoRETX setup, across all SINR levels and reference-based metrics.

*Table 10: Percentage improvement over NoRETX for BLEU, BLEURT, and BERTScore under Rayleigh conditions.*

| 2*SINR [dB] | BLEU | | BLEURT | | BERTScore | |
|---|---|---|---|---|---|---|
| | Baseline | Memory | Baseline | Memory | Baseline | Memory |
| -4 | 0.534% | 1.970% | 1.854% | 4.199% | 1.159% | 3.147% |
| 0 | 0.14% | 1.241% | 0.396% | 2.446% | 0.301% | 1.935% |
| 4 | 0.059% | 1.119% | 0.270% | 2.197% | 0.137% | 1.782% |
| 8 | 0.030% | 1.103% | 0.144% | 2.146% | 0.057% | 1.817% |
| 12 | 0.022% | 1.113% | 0.083% | 2.099% | 0.044% | 1.843% |
| 16 | 0.008% | 1.098% | 0.032% | 2.064% | 0.022% | 1.857% |
| 20 | 0.009% | 1.099% | 0.029% | 2.071% | 0.022% | 1.866% |

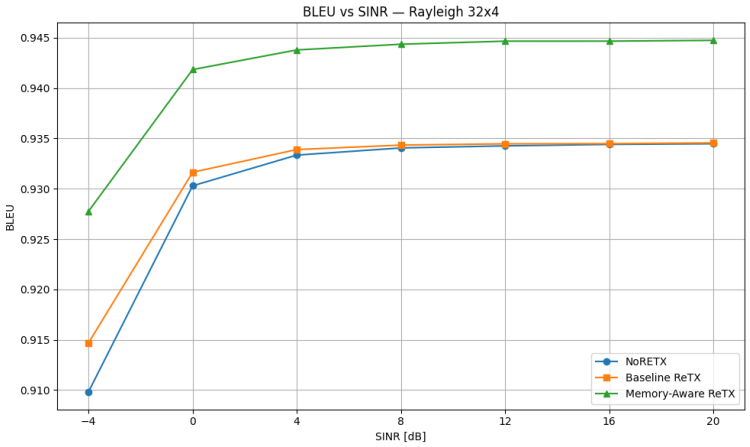



*Figure 29: BLEU scores under Rayleigh conditions comparing NoRETX, FullRETX (baseline), and FullRETX (memory-aware).*

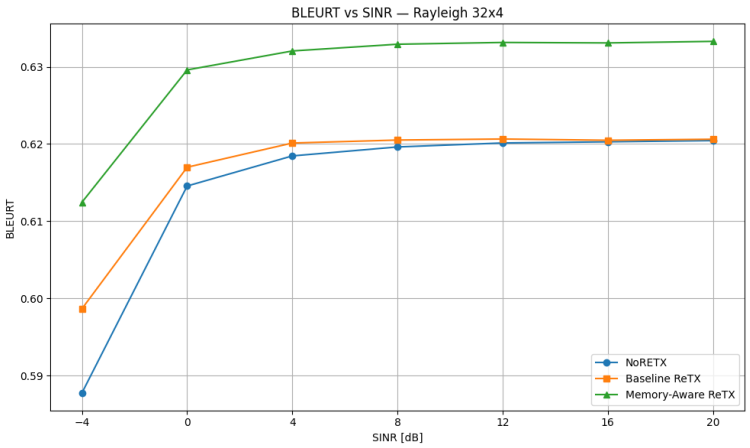



*Figure 30: BLEURT scores under Rayleigh conditions comparing NoRETX, FullRETX (baseline), and FullRETX (memory-aware).*

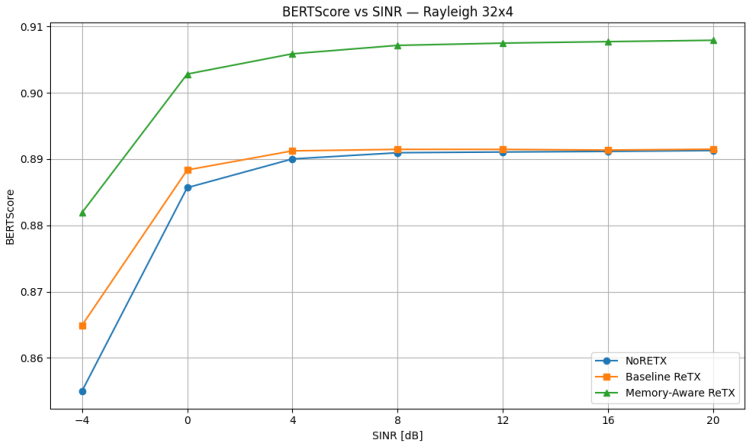



*Figure 31: BERTScore under Rayleigh conditions comparing NoRETX, FullRETX (baseline), and FullRETX (memory-aware).*

### Reference-Free Metrics

Figures 32, 33, and 34 report Perplexity (PPL), unigram-level Perplexity (PPLU), and CoLA acceptability under Rayleigh fading, comparing baseline and memory-aware configurations, both with and without retransmission.

As with CDL-B, lower values for PPL and PPLU indicate better performance, while higher values are preferred for CoLA.

In terms of PPL, the *baseline configuration* exhibits irregular behavior under retransmission. As shown in Table 11, no PPL values are available at low SINR levels (-4dB and 0dB), likely due to numerical instability or decoding failures caused by highly corrupted input, which make the perplexity score undefined. At moderate SINR (e.g., 8dB), retransmission leads to a peak improvement of **-2.56%**, but this trend reverses at higher SINRs: performance degrades slightly compared to the NoRETX setup, with PPL increasing by **+0.79%** at 16dB and **+0.15%** at 20dB. These inconsistencies are unexpected and suggest that the statistical relevance of the testing dataset should be improved by averaging results over a higher number of epochs — a process that may, however, be demanding due to the significant computational load involved.

In contrast, the *memory-aware configuration* shows strong and consistent PPL improvements across all SINR values (from 4dB onward). The gains are especially pronounced at higher SINRs, reaching a maximum reduction of **-15.23%** at 16dB, and remaining above **-10%** across the upper SINR range. These results confirm that memory mechanisms can be beneficial for effectively leveraging retransmission, enabling the system to refine its outputs over multiple decoding attempts, especially when the input signal is less noisy.

In terms of PPL, the *baseline configuration* also exhibits, in this case, inconsistent behavior under retransmission. As shown in Table 11, no values are available at low SINR levels (-4dB and 0dB), likely due to numerical instability or decoding failures under extremely noisy conditions. At moderate SINRs, retransmission yields mixed results: a modest improvement of **-2.56%** is observed at 8dB and **-2.21%** at 12dB, but these gains are offset by degradations at higher SINRs, such as **+0.79%** at 16dB and **+0.15%** at 20dB.

Surprisingly, the *memory-aware configuration* does not align with the expected trend of improved PPL through retransmission. Across all available SINR values, PPL increases when

retransmission is used. For example, the degradation reaches **+12.62%** at 4dB, **+15.23%** at 16dB, and **+14.22%** at 20dB, suggesting a systematic decline in performance. This behavior may result from overgeneration or instability introduced by repeated decoding steps, especially under favorable channel conditions, or, once again, from the limited number of simulated epochs used to average the results.

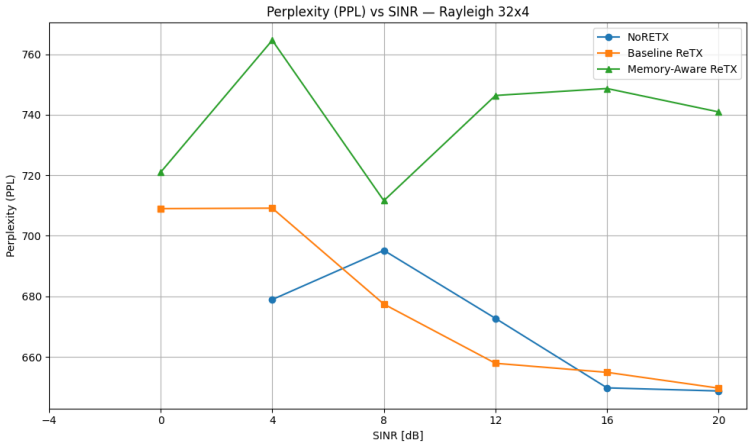



*Figure 32: PPL under Rayleigh conditions comparing NoRETX, FullRETX (baseline), and FullRETX (memory-aware). Lower values are better.*

For PPLU, which measures unigram-level token predictability, the differences across configurations are minimal. The *baseline configuration* shows very small variations from retransmission, with mostly negligible improvements. The best result is observed at -4dB, with a reduction of **-0.25%**, indicating slightly enhanced token-level consistency under poor channel conditions. However, the gains quickly diminish: at 0dB, the improvement drops to only **-0.009%**, and at 4dB a minor degradation of **+0.004%** is recorded. From this point onward, the trend slightly recovers, with small improvements up to **-0.006%** at 16dB. Overall, the effect of retransmission remains marginal in the absence of memory.

The *memory-aware setup*, by contrast, demonstrates more consistent behavior. Retransmission leads to slightly lower PPLU values across all SINRs, with improvements gradually increasing as channel conditions improve. From 0dB onward, the gap relative to the no-retransmission case widens, reaching a maximum reduction of **-0.14%** at 20dB. This behavior suggests that while memory-based retransmission provides only small benefits at the token level, it helps maintain local consistency more robustly, especially under favorable channel conditions.

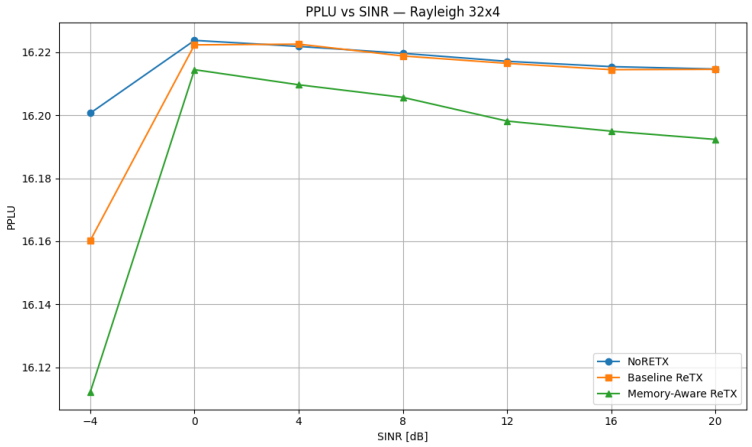



*Figure 33: PPLU under Rayleigh conditions comparing NoRETX, FullRETX (baseline), and FullRETX (memory-aware). Lower values are better.*

CoLA acceptability shows the *most consistent improvement* across all metrics and configurations. In the *baseline case*, retransmission yields a substantial gain of **+22.31%** at -4dB, with benefits decreasing steadily at higher SINRs, down to **+1.84%** at 20dB.

The *memory-aware configuration*, however, offers a more stable profile: improvements range from **+8.60%** to **+18.74%**, and maintain an upward trend as SINR increases. These results indicate that while the baseline benefits more at low SINR, particularly at -4dB where it achieves a higher gain than the memory-aware case (**+22.31%** vs. **+18.74%**), its performance deteriorates as channel conditions improve. In contrast, the memory-aware model continues to leverage retransmission effectively, preserving grammatical integrity at all SINR levels. Overall, the memory-enhanced system outperforms the baseline in most cases, providing more robust and reliable improvements in syntactic acceptability.

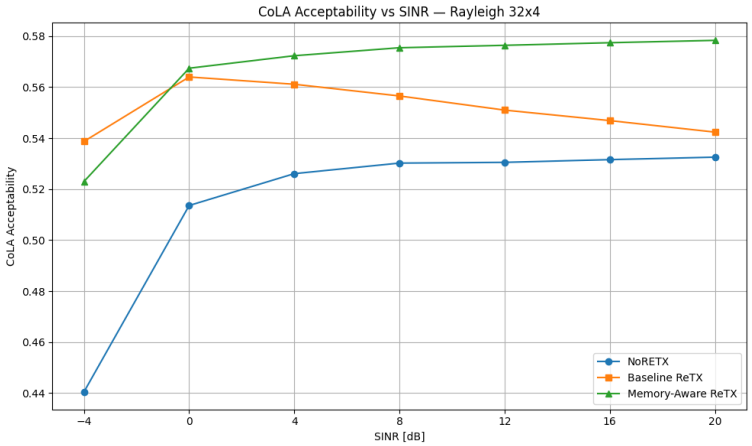



*Figure 34: CoLA acceptability under Rayleigh conditions comparing NoRETX, FullRETX (baseline), and FullRETX (memory-aware). Higher values are better.*

Table 11 quantifies these trends by reporting the percentage improvement of both baseline and memory-aware configurations over the NoRETX setup, across all SINR levels and reference-based metrics.

*Table 11: Percentage improvement over NoRETX for PPL, PPLU, and CoLA under Rayleigh conditions. For PPL and PPLU, lower values indicate better performance, so negative percentages imply improvement.*

| 2*SINR [dB] | PPL | | PPLU | | CoLA | |
|---|---|---|---|---|---|---|
| | Baseline | Memory | Baseline | Memory | Baseline | Memory |
| -4 | N/A | N/A | -0.249% | -0.547% | +22.307% | +18.738% |
| 0 | N/A | N/A | -0.009% | -0.057% | +9.819% | +10.488% |
| 4 | +4.450% | +12.623% | +0.004% | -0.075% | +6.665% | +8.793% |
| 8 | -2.555% | +2.363% | -0.005% | -0.086% | +4.967% | +8.535% |
| 12 | -2.206% | +10.953% | -0.004% | -0.117% | +3.865% | +8.656% |
| 16 | +0.789% | +15.228% | -0.006% | -0.126% | +2.881% | +8.629% |
| 20 | +0.145% | +14.218% | -0.001% | -0.138% | +1.843% | +8.604% |

## 4.3 Retransmission Behavior: Qualitative Analysis

To illustrate the practical effects of semantic retransmission logic, this section presents several decoding examples at fixed -4dB SNR under Rayleigh fading, using the baseline system with retransmission enabled. Each entry includes the transmitted and received sentence, the number of retransmission attempts, and key quality metrics. The selected cases demonstrate a variety of linguistic and semantic behaviors, including fluency, syntactic correctness, and semantic drift.

*Note: The received sentence and associated metrics correspond to the final decoding output, i.e., the version accepted after the last retransmission (if any).*

`Rayleigh, Baseline, ReTX enabled @ -4dB`

**Case A — No retransmission. Perfect reconstruction.**

Transmitted: `as a result and because we believe that this is an important report my group will vote in favour`

Received: `as a result and because we believe that this is an important report my group will vote in favour`

ReTX done: 0 — BLEU: 1.000, BLEURT: 0.739, BERT: 1.000, PPL: 68.6, PPLU: 19.00, CoLA: True

The sentence is recovered exactly, with all quality metrics confirming optimal decoding. No retransmission is needed.

**Case B — No retransmission. Fluent but syntactically flawed.**

Transmitted: `let us not forget that in the past four years georgia has made tremendous efforts to move forward toward a democratic and market oriented society`

Received: `let us not forget that in the past four years georgia has made tremendous efforts to move forward adopts a democratic and market roots society`

ReTX done: 0 — BLEU: 0.903, BLEURT: 0.586, BERT: 0.818, PPL: 149.3, PPLU: 25.00, CoLA: False

Despite grammatical issues and failed CoLA classification, the low PPL value leads to acceptance. This reflects a design choice in the ARQ logic that prioritizes fluency.

**Case C — One retransmission. Syntax improves after retry.**

Transmitted: `pl mr president in my contribution to the debate on the report on public finances in the emu countries for i would like to make a few observations`

Received: `pl mr president in my contribution to the debate on the report on the finances in the emu countries for i would like to make a few observations`

ReTX done: 1 — BLEU: 0.951, BLEURT: 0.706, BERT: 0.956, PPL: 125.2, PPLU:

19.80, CoLA: False

Retransmission is triggered once, and the final output remains fluent but not syntactically valid, confirming the conservative nature of the decision logic. Furthermore, the transmitted sentence appears syntactically incorrect, reducing the effectiveness of retransmission in restoring grammaticality.

**Case D — One retransmission. Improved output with positive CoLA.**

Transmitted: `it is true that russia is exploiting the situation but it is equally clear that russia s imperial interests would also find other justifications should the need arise`

Received: `it is true that russia is roots the situation but it is equally clear that russia s appointments interests would also find other adopts should the need arise`

ReTX done: 1 — BLEU: 0.820, BLEURT: 0.363, BERT: 0.702, PPL: 184.1, PPLU: 20.42, CoLA: True

Although PPL exceeds 150, the sentence passes CoLA and is thus accepted after a single retry. This example highlights the benefit of multi-criteria evaluation in moderate-noise regimes.

**Case E — Two retransmissions. Fluency dominates over syntax.**

Transmitted: `the stress tests should be evaluated on technical grounds and not on political grounds or grounds which leave room for speculation`

Received: `on stress tests should be fulfilled on technical grounds and not on political grounds or grounds which leave room for speculation`

ReTX done: 2 — BLEU: 0.908, BLEURT: 0.629, BERT: 0.874, PPL: 136.6, PPLU: 15.34, CoLA: False

Despite structural corruption at the start of the sentence and failed CoLA, PPL remains below threshold. As a result, the final decoding is accepted without further retries.

**Case F — Two retransmissions. CoLA enables semantic recovery.**

Transmitted: `i do not think that parliament is providing demonstrators or organising the demonstration therefore i cannot answer your question`

Received: `i do not think that parliament is providing demonstrators or preconditions the demonstration therefore i cannot answer your question`

ReTX done: 2 — BLEU: 0.909, BLEURT: 0.658, BERT: 0.809, PPL: 208.6, PPLU:

17.66, CoLA: True

Here, PPL is moderate, but CoLA classifies the output as acceptable. The logic correctly halts after the second attempt, avoiding unnecessary retransmissions.

**Case G — Three retransmissions. Poor result still accepted.**

Transmitted: `mr president mr barroso how nice it is for us to be able to continue our conversation that was so rudely interrupted just a month ago`

Received: `mr president mr barroso how nice it is for us to be able to continue our generate that was so midday down just a month ago`

ReTX done: 3 — BLEU: 0.796, BLEURT: 0.329, BERT: 0.687, PPL: 139.2, PPLU: 23.37, CoLA: False

The final output is both ungrammatical and semantically nonsensical. Nevertheless, the low PPL allows acceptance. This case exposes a limitation of threshold-only criteria.

**Case H — Three retransmissions. Semantic error, structurally valid.**

Transmitted: `thank you mrs leperre verrier i shall of course be sending parliament s condolences to the families concerned`

Received: `thank you mrs james blottnitz i shall of course be sending parliament s condolences to the families concerned`

ReTX done: 3 — BLEU: 0.868, BLEURT: 0.579, BERT: 0.705, PPL: 427.9, PPLU: 18.00, CoLA: True

A name substitution introduces a semantic distortion, yet the structure is fluent and passes CoLA. This illustrates how certain critical errors may go undetected under current policy.

**Case I — Four retransmissions. Retry budget exhausted despite poor output.**

Transmitted: `mr president this debate is becoming very emotional but my comments are not in that vein`

Received: `mr president this debate is becoming very jan but my comments are not in that phenomena`

ReTX done: 4 — BLEU: 0.862, BLEURT: 0.384, BERT: 0.743, PPL: 954.4, PPLU: 16.00, CoLA: False

Despite failing all linguistic checks (high perplexity, incorrect grammar, and semantic corruption), the sentence is accepted due to reaching the maximum number of retransmission attempts. This reveals a structural limit in the ARQ logic, which halts retrying

after three attempts regardless of output quality.

**Case J — Four retransmissions. Acceptable output after final attempt.**
Transmitted: `illegal immigration will also be tackled more effectively this involves a common repatriation policy and better border control`
Received: `illegal immigration will also be tackled more effectively this involves a common occurred policy and better border control`
ReTX done: 4 — BLEU: 0.908, BLEURT: 0.616, BERT: 0.852, PPL: 534.9, PPLU: 18.00, CoLA: True
Here, the system reaches the retry limit but produces a linguistically valid sentence. While a key noun is substituted, the output remains grammatical and coherent, leading to CoLA acceptance. This case demonstrates that, in some scenarios, retry budget exhaustion can still yield an acceptable decoding.

These examples demonstrate how the ARQ logic combines multiple reference-free metrics to guide retransmissions, balancing fluency (PPL), grammaticality (CoLA), and contextual coherence (PPLU). Most retransmissions contribute to output quality, although some edge cases reveal potential for improved semantic filtering beyond fixed thresholds.

# 5 Conclusion and Future Work

This thesis introduced a semantic communication system that extends the DeepSC framework with mechanisms aimed at improving performance in realistic transmission conditions. The proposed architecture integrates advanced semantic quality metrics, both reference-based (BLEU, BLEURT, BERTScore) and reference-free (PPL, PPLu, CoLA), and supports a retransmission strategy based on semantic acceptability. In particular, a memory-aware retransmission model was implemented, capable of combining information from multiple transmission attempts to enhance decoding quality.

The experimental analysis, conducted under CDL-B and Rayleigh fading channels, demonstrated that the use of memory can improve semantic reconstruction compared to both the no-retransmission case and traditional retransmission strategies without memory. These improvements, even if limited, have been observed across various evaluation metrics and signal-to-noise ratios, showing that the integration of prior communication attempts might be used to improve semantic robustness.

Despite the limited gains, this research direction remains worth pursuing, as several challenges are still open and point to potential improvements:

- **Improving semantic fidelity at high SNR levels.** Even in optimal channel conditions, reconstructed messages are not always perfect. This highlights the potential for improving the underlying encoder–decoder architecture or adopting more advanced language models. An improved architecture that can fully leverage the channel in favorable conditions is also more likely to benefit from a retransmission strategy, which leads to a more stable and higher-quality effective channel.

- **Extending support to multiple full retransmissions.** The current system allows only one enhanced retransmission. Future work may explore architectures capable of managing a longer sequence of retransmissions, each conditioned on previously received information.

- **Exploring adaptive encoder retransmission strategies.** In the current system, in the memory-aware setup, retransmissions are generated using a separate encoder network, resulting in a different latent representation from the original transmission. Future work

could expand this approach and explore the use of richer feedback signals to guide the construction of retransmissions, allowing the transmitter to adaptively refine or complement the original message. This could enhance semantic recovery by increasing diversity and reducing redundancy across retransmission attempts.

- **Leveraging semantic reconstructions instead of raw signals.** The current fusion strategy operates at the physical layer. An alternative approach would involve using the semantic output from a previous decoding attempt as the input to subsequent transmissions, shifting the focus from signal accumulation to meaning refinement.

- **Generalizing to other media types.** Although this work focused on natural language, the underlying methodology can be adapted to other modalities such as images, audio, or video, enabling semantic communication in a broader range of applications.

- **Evaluating on diverse and more ambiguous datasets.** The experiments in this thesis were conducted on the Europarl corpus, which contains formal and well-structured parliamentary language. Future evaluations on datasets with more informal, ambiguous, or conversational content (e.g., movie subtitles, dialogue datasets, or user-generated text) would provide deeper insight into the system's generalizability and semantic robustness in real-world scenarios.

- **Analyzing radio resource efficiency.** A detailed evaluation of the trade-offs between semantic performance and communication cost, in terms of bandwidth, energy, and latency, is essential for assessing the practical feasibility of such systems.

Semantic communication represents a novel and rapidly evolving research area where deep learning and communication theory intersect to redefine how information is transmitted and interpreted. The contributions developed in this thesis lay a foundation for future advancements in this direction, highlighting the importance of feedback, memory, and semantic reasoning in next-generation communication systems.

# References

[1] Samuel R. Bowman Alex Warstadt, Amanpreet Singh. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2019. URL: https://arxiv.org/abs/1805.12471.

[2] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, page 65–72, 2005. URL: https://aclanthology.org/W05-0909/.

[3] Benjamin Clavié Orion Weller Oskar Hallström Said Taghadouini Alexis Gallagher Raja Biswas Faisal Ladhak Tom Aarsen Nathan Cooper Griffin Adams Jeremy Howard Benjamin Warner, Antoine Chaffin and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. *arXiv preprint arXiv:2412.13663*, 2024. URL: https://arxiv.org/abs/2412.13663.

[4] Zhuoyuan Huo Khe Chai Sim Dongseong Hwang, Weiran Wang and Pedro Mengibar. Transformerfam: Feedback attention is working memory. *arXiv preprint arXiv:2404.09173*, 2024. URL: https://arxiv.org/abs/2404.09173.

[5] Sergio Barbarossa Seong-Lyun Kim Jinho Choi Eleonora Grassucci, Jihong Park and Danilo Comminiello. Generative ai meets semantic communication: Evolution and revolution of communication tasks. *arXiv preprint arXiv:2401.06803*, 2024. URL: https://arxiv.org/abs/2401.06803.

[6] Google Cloud Translation. Informazioni sul punteggio bleu, 2024. URL: https://cloud.google.com/translate/automl/docs/evaluate#bleu_score.

[7] Antonio Pio Grieco. Simulation analysis of semantic communications applied to a human language use case. Master's thesis, Politecnico di Torino, 2024. Supervisor: Prof. Carla Fabiana Chiasserini.

[8] Yunlong Cai Guangyi Zhang, Qiyu Hu and Guanding Yu. Scan: Semantic communication with adaptive channel feedback. *arXiv preprint arXiv:2306.15534*, 2023. URL: https://arxiv.org/abs/2306.15534.

[9] Michael Hanna and Ondrej Bojar. A fine-grained analysis of bertscore. In *Proceedings of the Sixth Conference on Machine Translation*, 2021. URL: https://aclanthology.org/2021.wmt-1.59/.

[10] Geoffrey Ye Li Huiqiang Xie, Zhijin Qin and Biing-Hwang Juang. Deep learning enabled semantic communication systems. *arXiv preprint arXiv:2006.10685*, 2021. URL: https://arxiv.org/abs/2006.10685.

[11] Zhijin Qin Huiqiang Xie and Geoffrey Ye Li. Semantic communication with memory. *arXiv preprint arXiv:2303.12335*, 2023. URL: https://arxiv.org/abs/2303.12335.

[12] Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2019. URL: https://arxiv.org/abs/1810.04805.

[13] Zhen Qi Zhenhong Zhang Chihang Wang Jiajing Chen, Shuo Wang and Hongye Zheng. A combined encoder and transformer approach for coherent and high-quality text generation. *arXiv preprint arXiv:2411.12157*, 2024. URL: https://arxiv.org/abs/2411.12157.

[14] Wenjun Xu Hui Gao Jiangjing Hu, Fengyu Wang and Ping Zhang. Semharq: Semantic-aware harq for multi-task semantic communications. *arXiv preprint arXiv:2404.08490*, 2024. URL: https://arxiv.org/abs/2404.08490.

[15] Chuan Huang Jianhao Huang, Kai Yuan and Kaibin Huang. $D^2$-jscc: Digital deep joint source-channel coding for semantic communications. *arXiv preprint arXiv:2403.07338*, 2024. URL: https://arxiv.org/abs/2403.07338.

[16] Zehui Xiong Chengwen Xing Rahim Tafazolli Tony Q.S. Quek Fellow IEEE Jiayi Lu, Wanting Yang and Merouane Debbah. Generative ai-enhanced multi-modal seman-

tic communication in internet of vehicles: System design and methodologies. *arXiv preprint arXiv:2409.15642*, 2024. URL: https://arxiv.org/abs/2409.15642.

[17] Soo-Young Lee Jihyeon Roh, Sang-Hoon Oh. Unigram-normalized perplexity as a language model performance measure with different vocabulary sizes. *arXiv preprint arXiv:2011.13220*, 2020. URL: https://arxiv.org/abs/2011.13220.

[18] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

[19] Wensheng Lin-Yuna Yan Rui Li Wenchi Cheng Kexin Zhang, Lixin Li and Zhu Han. Semantic successive refinement: A generative ai-aided semantic communication framework. *arXiv preprint arXiv:2408.05112*, 2024. URL: https://arxiv.org/abs/2408.05112.

[20] Todd Ward Kishore Papineni, Salim Roukos and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, 2002. URL: https://aclanthology.org/P02-1040/.

[21] Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, 2007. URL: https://aclanthology.org/W07-0734/.

[22] Pyae Sone Aung Dusit Niyato Zhu Han Loc X. Nguyen, Avi Deb Raha and Choong Seon Hong. A contemporary survey on semantic communications: Theory of mind, generative ai, and deep joint source-channel coding. *arXiv preprint arXiv:2502.16468*, 2025. URL: https://arxiv.org/abs/2502.16468.

[23] Guangxu Zhu Richeng Jin Xiaoming Chen Maojun Zhang, Haotian Wu and Deniz Gunduz. Semantics-guided diffusion for deep joint source-channel coding in wireless image transmission. *arXiv preprint arXiv:2501.01138*, 2025. URL: https://arxiv.org/abs/2501.01138.

[24] Clara Meister and Ryan Cotterell. Language model evaluation beyond perplexity. *arXiv preprint arXiv:2106.00085*, 2021. URL: https://arxiv.org/abs/2106.00085.

[25] Baptiste Colle Mircea Lica, Ojas Shirekar and Chirag Raman. Mindforge: Empowering embodied agents with theory of mind for lifelong collaborative learning. *arXiv preprint arXiv:2411.12977*, 2025. URL: https://arxiv.org/abs/2411.12977.

[26] Kai Niu and Ping Zhang. A mathematical theory of semantic communication. *arXiv preprint arXiv:2401.13387*, 2024. URL: https://arxiv.org/abs/2401.13387.

[27] Matthew Raffel and Lizhong Chen. Implicit memory transformer for computationally efficient simultaneous speech translation. *arXiv preprint arXiv:2307.01381*, 2023. URL: https://arxiv.org/abs/2307.01381.

[28] Hadeel Saadany and Constantin Orăsan. Bleu, meteor, bertscore: Evaluation of metrics performance in assessing critical translation errors in sentiment-oriented text. In *arXiv preprint arXiv:2109.14250*, 2021. URL: https://arxiv.org/abs/2109.14250.

[29] Sarah Lee. Ultimate guide to meteor score in computational linguistics, 2024. URL: https://www.numberanalytics.com/blog/ultimate-guide-to-meteor-score-in-computational-linguistics.

[30] Yo-Seb Jeon Namyoon Lee Taewoo Park, Eunhye Hong and Yongjune Kim. Robust deep joint source channel coding for task-oriented semantic communications. *arXiv preprint arXiv:2503.12907*, 2025. URL: https://arxiv.org/abs/2503.12907.

[31] Kees van Deemter Takumi Ito and Jun Suzuki. Reference-free evaluation metrics for text generation: A survey. *arXiv preprint arXiv:2501.12011*, 2025. URL: https://arxiv.org/abs/2501.12011.

[32] Dipanjan Das Thibault Sellam and Ankur P. Parikh. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*, 2020. URL: https://arxiv.org/abs/2004.04696.

[33] Christo Kurisummoottil Thomas, Emilio Calvanese Strinati, and Walid Saad. Reasoning with the theory of mind for pragmatic semantic communication. *arXiv preprint arXiv:2311.18224*, 2023. URL: https://arxiv.org/abs/2311.18224.

[34] Felix Wu Kilian Q. Weinberger Tianyi Zhang, Varsha Kishore and Yoav Artz. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2020. URL: https://arxiv.org/abs/1904.09675.

[35] Meixia Tao Xiaodong Xu Wenjun Zhang Tong Wu, Zhiyong Chen and Ping Zhang. Mambajscc: Deep joint source-channel coding with visual state space model. *arXiv preprint arXiv:2405.03125*, 2024. URL: https://arxiv.org/abs/2405.03125.

[36] Julien CHAUMOND Victor SANH, Lysandre DEBUT and Thomas WOLF. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2020. URL: https://arxiv.org/abs/1910.01108.

[37] W. Weaver. Recent contributions to the mathematical theory of communication. *ETC: A Review of General Semantics*, pages 261–281, 1949. URL: https://waste.informatik.hu-berlin.de/Lehre/ss11/SE_Kybernetik/reader/weaver.pdf.

[38] Hao Ye Shi Jin Yucheng Sheng, Le Liang and Geoffrey Ye Li. Semantic communication for cooperative perception using harq. *arXiv preprint arXiv:2409.09042*, 2024. URL: https://arxiv.org/abs/2409.09042.

[39] Kartik Sreenivasan1 Max Marion Matthew L. Leavitt Mansheej Paul Zachary Ankner, Cody Blakeney. Perplexed by perplexity: Perplexity-based data pruning with small reference models. *arXiv preprint arXiv:2405.20541*, 2024. URL: https://arxiv.org/abs/2405.20541.