SENSE: Shared Embeddings for Naturalistic Sensing and Episodic Memory

Yann Guszkiewicz with the assistance of certain LLM
Independent Researcher
yannougus@gmail.com

Abstract

Recent advances in large language models (LLMs) have demonstrated remarkable abilities in natural language understanding and reasoning, yet most remain fundamentally text-bound. While multimodal extensions exist, they typically process each sensory modality—such as vision or audio—through independent encoders that translate inputs into intermediate textual descriptions or modality-specific embeddings before integration. This approach often limits cross-modal alignment and hinders continuous, real-time multimodal perception.

We propose ***SENSE*** *(Shared Embeddings for Naturalistic Sensing and Episodic memory), a novel architecture in which modality-specific models (vision, audio, and text) share a unified tokenizer and thus operate within a single, consistent embedding space. This shared token space enables direct semantic alignment between sensory inputs—e.g., an image of a car, the sound of its engine, and the textual concept "car"—without lossy translation to intermediate textual form.*

To address the limited context windows of current LLMs, we introduce mini-RAG modules: lightweight, distributed retrieval-augmented memories that store continuous streams of multimodal tokens as temporally segmented episodes. The central LLM selectively queries these episodic memories based on semantic relevance, enabling virtually unbounded temporal context without overwhelming the model's input length.

We outline the architecture, training objectives, and potential deployment strategies of SENSE, discuss its alignment with existing multimodal research, and highlight its potential for real-time, scalable, and privacy-conscious perceptual AI systems.

Contents

1 Introduction

Large language models (LLMs) have redefined the landscape of artificial intelligence, excelling in tasks ranging from machine translation to complex reasoning. However, even the most advanced LLMs, including multimodal variants, lack the seamless, continuous integration of sensory inputs that characterizes human perception. Existing multimodal systems often employ modality-specific pipelines, where vision or audio data are first processed into textual descriptions or mapped to embeddings that are later fused. While effective for discrete tasks, this architecture can fragment semantic alignment and limit the capacity for long-term, real-time understanding of an environment.

Motivation. Human cognition thrives on an interconnected sensory experience: the sight of a phone, the sound it makes when placed on a table, and the linguistic label "phone" are bound together in a shared conceptual space. Inspired by this, we propose an architecture in which all modalities communicate through a *shared tokenizer*—a unified vocabulary of tokens representing concepts agnostic to their origin. This shared space not only facilitates natural cross-modal alignment but also supports retrieval and reasoning across heterogeneous sensory histories.

Challenges. A key limitation of LLMs is the finite *context window*. Continuous sensory streams quickly exceed this limit, making it impossible to retain a detailed episodic record over time. Moreover, naively fusing modalities often increases input redundancy and computational overhead. Finally, aligning vision, audio, and text at the token level presents nontrivial training and representation learning challenges.

Our Approach. We introduce **SENSE**, an architecture that unifies three modality-specific encoders—for vision, audio, and text—under a single tokenizer and embedding space. To manage temporal scale, SENSE incorporates *mini-RAG* modules: distributed retrieval-augmented memories that segment and store multimodal token streams as discrete episodes. Each mini-RAG maintains its own lightweight language model for local indexing and relevance estimation. When queried by the central LLM, only the most relevant segments are retrieved and integrated, effectively extending the usable temporal context to an unbounded horizon.

Contributions. Our main contributions are:

1. A unified multimodal tokenization framework enabling vision, audio, and language models to operate in a single embedding space.
2. A distributed, continuous episodic memory architecture (*mini-RAG*) that scales temporal context without inflating the central LLM's input window.
3. An attention-based retrieval mechanism allowing the central LLM to correlate and weight multimodal evidence for contextual reasoning.
4. A discussion of potential deployment scenarios, datasets, training regimes, and privacy considerations for real-time multimodal agents.

In the following sections, we position SENSE in the context of prior multimodal and retrieval-augmented research, detail its architecture and training strategy, and outline experiments to evaluate its performance across a range of retrieval and reasoning tasks.

2 Related Work

Multimodal Representation Learning. A large body of work has sought to align visual and textual modalities in a shared embedding space, most notably CLIP [1], which employs contrastive learning over image–text pairs. More recent efforts such as ImageBind [2] extend this paradigm to additional modalities (e.g., audio, depth, thermal) using modality-specific encoders trained to produce aligned embeddings. While these models demonstrate strong cross-modal retrieval performance, they rely on independent tokenization and late fusion, which limits their capacity for token-level cross-modal reasoning.

Multimodal Large Language Models. Models such as Kosmos-1 [3], Flamingo [4], and Gato [5] integrate multimodal inputs into a unified transformer backbone, typically by projecting non-textual modalities into a textual embedding space. Although these approaches enable a degree of cross-modal understanding, their reliance on intermediate textualization or modality-specific projection layers introduces potential semantic loss and hinders continuous, real-time integration.

Retrieval-Augmented Generation and Episodic Memory. Retrieval-augmented generation (RAG) [6] augments language models with external memory for grounding and factual accuracy. Extensions of this paradigm to multimodal domains [7] typically store and retrieve embeddings from a centralized database. Episodic memory architectures [8] aim to maintain temporally structured records of interactions, but often face scalability challenges when deployed in continuous, high-bandwidth sensory contexts.

Positioning of SENSE. SENSE draws inspiration from these prior works but differs in three key aspects: (1) the use of a *shared tokenizer* across all modalities, enabling token-level cross-modal alignment without intermediate textual bottlenecks; (2) the integration of *distributed mini-RAG modules* that store temporally segmented multimodal tokens locally; and (3) a retrieval mechanism allowing a central LLM to selectively attend to and correlate multimodal evidence across distributed episodic memories. To our knowledge, no prior architecture combines these features into a unified framework for scalable, real-time multimodal reasoning.

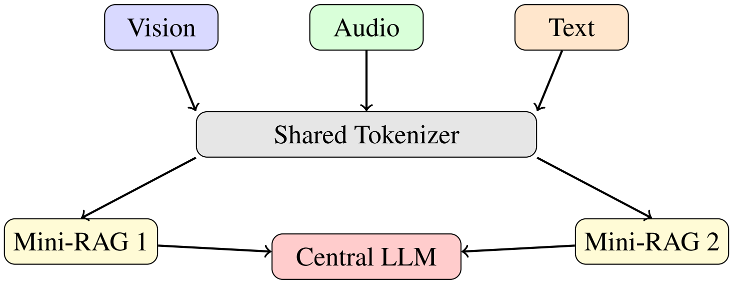

Figure 1: Overview of the SENSE architecture. Modality-specific encoders (vision, audio, text) feed into a *shared multimodal tokenizer*, producing a unified token stream. Tokens are segmented and stored in distributed *mini-RAG* nodes, which can be queried by a central LLM equipped with cross-modal attention for selective reasoning.

3 Model Architecture

This section formalizes the SENSE architecture. We introduce notation, present two concrete designs for a shared multimodal tokenizer, describe modality encoders, detail the mini-RAG node specification, and give the central LLM retrieval + cross-modal attention mechanism. We then summarize the complete query-time algorithm.

3.1 Notation

Let $\mathcal{V}, \mathcal{A}, \mathcal{T}$ denote the spaces of raw vision (images / video), audio, and text inputs respectively. We denote modality-specific encoders:

$$E_v : \mathcal{V} \to \mathbb{R}^{n_v \times d}, \qquad E_a : \mathcal{A} \to \mathbb{R}^{n_a \times d}, \qquad E_t : \mathcal{T} \to \mathbb{R}^{n_t \times d},$$

where each encoder outputs a sequence of n_* vectors in a shared embedding dimension d. The central object is a *shared tokenizer* $T(\cdot)$ which maps encoder outputs to sequences of discrete or continuous *tokens* in a common token space:

$$T : \mathbb{R}^{n \times d} \to \{z_1, \ldots, z_m\}, \qquad z_i \in \mathcal{Z},$$

where $\mathcal{Z}$ is either a discrete codebook or a continuous token manifold (see designs below).

We denote a *mini-RAG* node as $\mathcal{M}_i$, which stores a temporally ordered list of episodes $\{E_1^i, E_2^i, \ldots\}$, each episode represented as a compact token summary (sequence) and indexed in a vector database via keys in $\mathbb{R}^d$. The central LLM is denoted by $\mathcal{L}$; it accepts textual and token streams and supports cross-attention to retrieved episodes.

3.2 Shared Multimodal Tokenizer: two design options

We present two practical designs for $T(\cdot)$.

Design A: Continuous shared embedding with modality tags. Each encoder projects into the same d-dimensional space; tokens are produced by a lightweight segmentation operator that groups consecutive embeddings into chunks and represents each chunk by its mean vector. Formally, for encoder output $X \in \mathbb{R}^{n\times d}$, a segmentation $\mathcal{S} = \{I_1, \ldots, I_m\}$ partitions indices $\{1, \ldots, n\}$ and

$$z_j = \frac{1}{|I_j|} \sum_{t\in I_j} X_t, \qquad j = 1 \ldots m.$$

Each z_j is optionally concatenated with a one-hot modality tag $u \in \{e_v, e_a, e_t\}$ and passed through a projection head $P(\cdot)$ yielding token vectors in $\mathbb{R}^d$ used downstream.

Design B: Discrete codebook tokenizer (quantized tokens). We learn a shared codebook $\mathcal{C} = \{c_1, \ldots, c_K\} \subset \mathbb{R}^d$. Each encoder output vector x_t is quantized to the nearest code $q(x_t) = \arg\min_{c\in\mathcal{C}} \|x_t - c\|$. Tokens are sequences of code indices. Codebook learning can follow standard VQ or product quantization methods and is trained jointly with alignment losses (see Sec. 4).

Design A is simple and preserves continuous similarity structure; Design B yields compact discrete representations that are convenient for storage and token-budget accounting. Both can be augmented with positional/time encodings.

3.3 Modality Encoders

Encoders E_v, E_a, E_t can be implemented using state-of-the-art architectures adapted to produce d-dimensional vectors per local patch / time-step:

- Vision: patch-based ViT encoder producing token-per-patch features (optionally optical-flow encoders for motion tokens).
- Audio: short-time spectrogram backbone (e.g., convolutional transformer) yielding frame-wise embeddings.
- Text: standard subword tokenizer followed by a transformer embedding.

All encoders are trained so their outputs are aligned in the shared space via cross-modal contrastive objectives.

3.4 Mini-RAG Node

Each mini-RAG $\mathcal{M}_i$ receives a continuous stream of tokens $[z_1, z_2, \ldots]$ (from any modality) and executes the following pipeline:

1. **Segmentation**: group tokens into episodes using fixed windows, event-triggered boundaries, or learned segmentation signals (see Alg. 1).
2. **Local summarization**: condense each episode into a compact sequence of summary tokens via a lightweight summarizer (mini-LLM or sequence compressor).
3. **Indexing**: compute an episode-level key vector (e.g., mean of summary token embeddings) and insert into an approximate nearest neighbor (ANN) index for retrieval.
4. **Retention policy**: optionally apply retention rules (e.g., LRU, relevance-based retention, or privacy-driven purging).

Each $\mathcal{M}_i$ may run on-device or on a nearby edge server. The mini-LLM inside $\mathcal{M}_i$ is only used for compressive summarization and lightweight relevance scoring; it can be small (millions to low hundreds of millions of parameters) and quantized.

Algorithm 1 Episode Segmentation and Indexing (mini-RAG)

```
Require: token stream $\mathcal{Z} = \{z_t\}$, window size $W$, event detector $D(\cdot)$
buffer ← []
for each incoming token $z_t$ do
    append $z_t$ to $buffer$
    if $|buffer| \geq W$ or $D(buffer) =$ true then
        $episode \leftarrow buffer$
        $summary \leftarrow \mathsf{Summarize}(episode)$          ▷ mini-LLM compressor
        $key \leftarrow \mathsf{Key}(summary)$
        insert $(key, summary, metadata)$ into ANN index
        $buffer \leftarrow []$
    end if
end for
```

Algorithm 2 Query-Time Retrieval and Answering

```
Require: query $q$, mini-RAG set $\{\mathcal{M}_i\}$, coarse-K, rerank-k, token budget $B$
$Q \leftarrow \mathsf{EncQuery}(q)$
parallel retrieve from each $\mathcal{M}_i$: candidates $\mathcal{C}_i \leftarrow \mathsf{ANN\_search}(\mathcal{M}_i, Q,$ coarse-K$)$
$\mathcal{C} \leftarrow \bigcup_i \mathcal{C}_i$
for each candidate $c \in \mathcal{C}$ do
    compute $s_c \leftarrow \mathsf{CrossAttnScore}(Q, \mathsf{summary}(c))$
end for
select top-rerank-k candidates under budget $B$ (by $s_c$)
$\mathbf{H}_r \leftarrow$ concatenate summary tokens from selected candidates
$\mathsf{Answer} \leftarrow \mathcal{L}(q; \mathbf{H}_r)$          ▷ LLM generation conditioned on retrieved context
return Answer
```

3.5 Central LLM and Cross-Modal Attention

At query time, the central LLM $\mathcal{L}$ receives a user query q (natural language) and performs a selective retrieval from the distributed mini-RAGs. Retrieval operates in two stages:

1. **Coarse retrieval**: using q (embedded into $\mathbb{R}^d$) retrieve top-K candidate episodes from ANN across mini-RAGs (this can be parallelized).

2. **Cross-modal reranking**: for each candidate episode, run a small cross-attention module between the LLM's query representation and the episode summary tokens to compute a relevance score; select top-k episodes under a global token budget B.

Let $Q = \mathsf{EncQuery}(q) \in \mathbb{R}^d$ be the query embedding. Candidate keys $\{k_j\}$ are retrieved by nearest neighbor search. Reranking score for episode j:

$$s_j = \mathsf{CrossAttnScore}(Q, \{s_{j,1}, \ldots, s_{j,m_j}\}),$$

where $s_{j,\ell}$ are summary tokens for episode j. The central LLM then attends jointly over the retrieved summary tokens and the query, allowing the self-/cross-attention layers to correlate signals (e.g., visual token for "table" with audio token for "object placed").

Formally, if $\mathbf{H}_q$ are the query token embeddings and $\mathbf{H}_r$ are the concatenated retrieved summary token embeddings, a multi-head attention (MHA) block computes:

$$\mathrm{MHA}(\mathbf{H}_q, \mathbf{H}_r) = \mathrm{softmax}\left(\frac{\mathbf{H}_q W_Q (\mathbf{H}_r W_K)^\top}{\sqrt{d_h}}\right) \mathbf{H}_r W_V,$$

allowing the model to reweight retrieved multimodal evidence for downstream decoding.

3.6 Vision-specific compression: motion and persistence tokens

To reduce redundancy between successive frames, the vision encoder can emit two types of tokens:

- **Persistence tokens**: represent objects that persist across frames (tracked via appearance matching).
- **Motion tokens**: represent dynamic changes (computed via optical flow or frame difference).

By storing mainly motion tokens and occasional persistence snapshots, mini-RAG storage and retrieval efficiency improve while preserving the capacity to reason about object placement and actions.

3.7 Deployment considerations (brief)

SENSE supports multiple deployment modes:

- **On-device mini-RAGs**: privacy-preserving, low-latency local retrieval; periodic sync to cloud.
- **Edge-hosted mini-RAGs**: offload compute to a nearby gateway; good tradeoff for wearable devices.
- **Centralized mini-RAGs**: all memories stored in cloud—simpler but higher privacy/cost concerns.

In the next section we present the learning objectives used to align modalities, train the tokenizer, and ensure robust retrieval and continual learning.

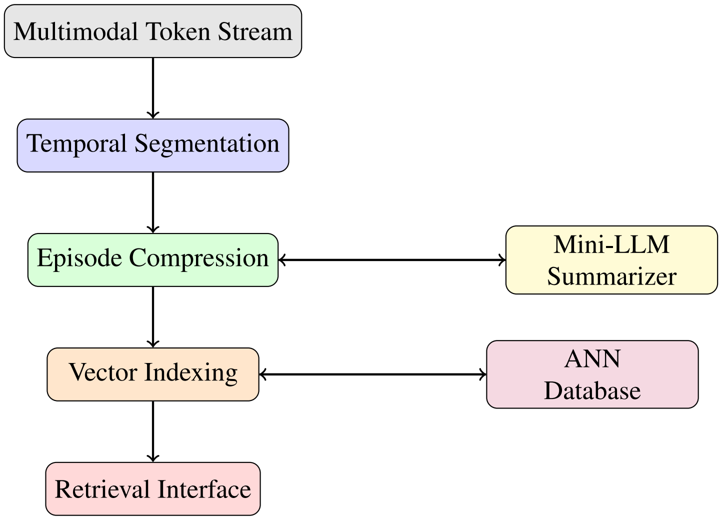

Figure 2: Internal pipeline of a *mini-RAG* node. Incoming multimodal tokens are segmented into episodes, summarized locally, assigned a key vector, indexed in an ANN structure, and optionally pruned via a retention policy.

4 Training Objectives

The SENSE architecture is trained to align multimodal representations, compress episodes effectively, and optimize retrieval performance. Training proceeds in stages, but all components are eventually fine-tuned jointly. We detail the key objectives below.

4.1 Cross-Modal Alignment Loss

To align vision, audio, and text tokens into a shared embedding space, we use a symmetric InfoNCE contrastive loss. Let $\mathbf{v}$, $\mathbf{a}$, and $\mathbf{t}$ denote pooled embeddings of corresponding clips. The loss for vision–text alignment is:

$$\mathcal{L}_{VT} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(\mathrm{sim}(\mathbf{v}_i,\mathbf{t}_i)/\tau)}{\sum_{j=1}^{N}\exp(\mathrm{sim}(\mathbf{v}_i,\mathbf{t}_j)/\tau)},$$

where sim is cosine similarity and τ a temperature parameter. Similar terms $\mathcal{L}_{AT}$ and $\mathcal{L}_{VA}$ are included.

4.2 Token Reconstruction Loss

When using discrete tokens (Design B), we add a vector-quantization commitment and reconstruction loss:

$$\mathcal{L}_{rec} = \|x - \hat{x}\|_2^2 + \beta \|\mathrm{sg}[x] - z_q\|_2^2,$$

where z_q is the quantized code and sg is the stop-gradient operator.

4.3 Episode Summarization Loss

The local summarizer inside each mini-RAG is trained with a compressive sequence-to-sequence objective. Given a full episode token sequence $\mathbf{Z}$ and a compressed summary $\hat{\mathbf{Z}}$, the summarizer is optimized to reconstruct salient information for downstream retrieval:

$$\mathcal{L}_{sum} = \mathrm{CE}\big(\mathcal{L}(\mathbf{Q}, \hat{\mathbf{Z}}), \mathcal{L}(\mathbf{Q}, \mathbf{Z})\big),$$

where $\mathcal{L}(\cdot, \cdot)$ denotes the central LLM's output distribution given query $\mathbf{Q}$ and context.

4.4 Retrieval and Reranking Loss

We optimize retrieval quality via a max-margin ranking loss. For query embedding q, positive key k^+, and negative keys $\{k^-\}$:

$$\mathcal{L}_{ret} = \sum_{k^-} \max\big(0, \gamma - \mathrm{sim}(q, k^+) + \mathrm{sim}(q, k^-)\big),$$

where γ is the margin.

4.5 Continual Learning Regularization

To mitigate catastrophic drift in mini-RAG summarizers, we apply:

- **Elastic Weight Consolidation (EWC)**: penalizing changes to parameters important for past tasks.
- **Replay buffer**: occasional re-training on a small set of past episodes.

This ensures that as the model updates with new sensory data, it retains competence on older events.

4.6 Joint Objective

The final loss is:

$$\mathcal{L} = \lambda_{align}\mathcal{L}_{align} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{sum}\mathcal{L}_{sum} + \lambda_{ret}\mathcal{L}_{ret} + \lambda_{cl}\mathcal{L}_{CL},$$

with weights λ_* tuned empirically.

5 Experimental Setup

We describe datasets, tasks, baselines, metrics, and ablation protocols to evaluate SENSE. The goal is to measure cross-modal alignment quality, retrieval accuracy in episodic settings, temporal reasoning capabilities, and practical deployment trade-offs (latency, storage, privacy).

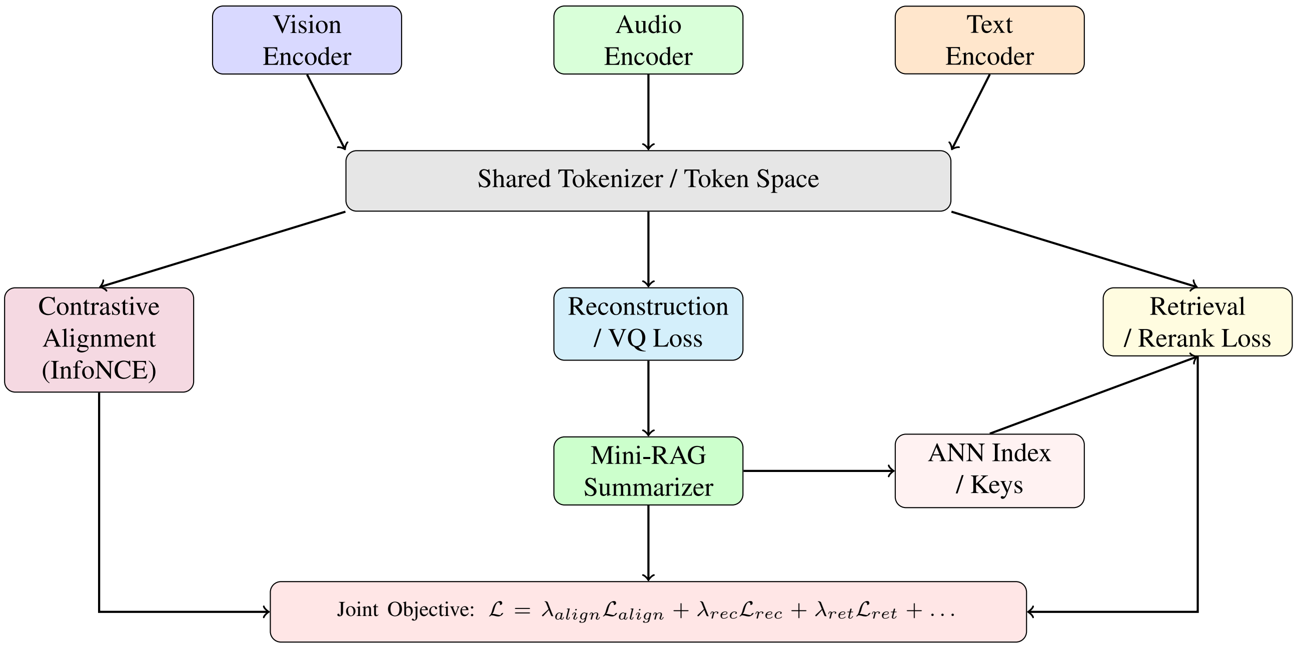
Vision Encoder
Audio Encoder
Text Encoder
Shared Tokenizer / Token Space
Contrastive Alignment (InfoNCE)
Reconstruction / VQ Loss
Retrieval / Rerank Loss
Mini-RAG Summarizer
ANN Index / Keys
Joint Objective: $\mathcal{L} = \lambda_{align}\mathcal{L}_{align} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{ret}\mathcal{L}_{ret} + \dots$

Figure 3: Training flows for SENSE. ...

5.1 Datasets and Data Collection

We consider a mix of large-scale, publicly available multimodal corpora and task-specific collections:

- **YouTube / Web-scale video corpora** (raw source): diverse, noisy, and abundant, suitable for large-scale pretraining when licensed and filtered appropriately.
- **HowTo100M / HowTo100M-like** (instructional videos): long-form synchronized audio/video pairs with spoken narrations useful for audio–video–text alignment.
- **Ego4D**: first-person video dataset with dense temporal annotations; useful for egocentric episodic tasks.
- **AudioSet / VGGSound / AVSpeech**: large audio-visual datasets for sound–video alignment and audio event detection.
- **YouCook2 / COIN / ActivityNet**: datasets with procedural activities, supporting temporal localization and action reasoning.

Data curation and ethics. For datasets sourced from web platforms, we will follow applicable terms of service and consider privacy: filter personally identifiable information, obtain licenses where necessary, and include a section detailing dataset provenance and consent considerations.

5.2 Evaluation Tasks

To comprehensively evaluate SENSE we propose the following tasks:

1. **Cross-modal retrieval**: given a query in one modality (text/audio/image), retrieve the corresponding segment in another modality. Metrics: Recall@K, MRR, mean average precision (mAP).
2. **Episodic retrieval / “Where did I put X?”**: temporal retrieval where the system must find the episode in which an object or event occurred (e.g., “Where did I put my phone?”). Metrics: retrieval accuracy, time-to-first-correct (latency), and precision@1 within a time window.

3. **Temporal QA**: question-answering about past events using retrieved multimodal context (e.g., "What object made the knocking sound at 10:23?"). Metrics: exact match / F1 on answers, human evaluation for ambiguous cases.
4. **Localization and tracking**: detect and localize objects across time (persistence tokens). Metrics: IoU for bounding boxes, tracking accuracy.
5. **Latency and resource tests**: measure end-to-end query latency under different mini-RAG placements (on-device, edge, cloud) and varying retrieval budgets. Metrics: average latency, p95 latency, bandwidth used, storage per hour of recorded stream.

5.3 Baselines

We include several baselines to situate performance:

- **ASR → LLM pipeline**: standard approach where audio is transcribed, vision is captioned (off-the-shelf), and an LLM consumes the text-only stream.
- **CLIP + RAG**: CLIP (image–text) for visual embeddings, combined with a RAG pipeline storing textual captions or CLIP embeddings in a vector DB, then queried by an LLM.
- **ImageBind-like embedding + RAG**: a joint embedding model (vision, audio, text) but with separate tokenization; episodes are indexed by pooled embeddings and retrieved for an LLM.
- **Unified multimodal transformer (Kosmos/Gato-style)**: a single transformer ingesting modality-specific projections as tokens (where available).

The purpose of these baselines is not adversarial competition but to highlight where SENSE's design choices lead to improved episodic retrieval or scaling properties.

5.4 Metrics

Core quantitative metrics include:

- Retrieval: Recall@K (K=1,5,10), MRR, mAP
- QA: Exact Match / F1 (for extractive answers), human-rated coherence/accuracy for generative answers
- Localization: IoU, tracking F1
- Efficiency: latency (mean, p95), bandwidth consumed, storage per hour, token budget usage for central LLM
- Robustness: performance under noise (audio SNR variations, visual occlusion), measured as degradation from clean performance
- Continual learning stability: retention (performance on a held-out old set after online updates), forgetting metrics

5.5 Ablation Studies

We propose systematic ablations to quantify the contribution of key design decisions:

- **Tokenizer design**: Design A (continuous + modality tags) vs Design B (discrete codebook). Measure retrieval and storage efficiency trade-offs.
- **Segmentation policy**: fixed-window vs event-triggered vs learned segmentation.
- **Summarizer capacity**: mini-LLM sizes (tiny → small → medium) and their effect on summary quality and retrieval.
- **Mini-RAG placement**: on-device vs edge vs cloud — measure latency, privacy, and bandwidth.

- **Retention policies**: LRU vs relevance-based vs privacy-driven purge, impact on long-term retrieval.
- **Cross-attention budget**: vary token budget B for central LLM and observe QA/retrieval trade-offs.
- **Continual learning**: with vs without EWC / replay; measure forgetting on older episodes.

5.6 Training and Compute Protocol

- **Pretraining stage**: large-scale contrastive alignment on web-scale video + audio + text (self-supervised), training encoders and tokenizer heads.
- **Summarizer / mini-RAG pretrain**: train summarizers with compressive seq2seq and retrieval objectives (synthetic queries from transcripts or automatically generated event prompts).
- **Joint fine-tuning**: fine-tune end-to-end on downstream tasks (retrieval, QA) while applying continual learning regularizers.

We will report common hyperparameters (batch size, learning rates, optimizer, number of steps), and when reporting compute cost provide an estimate of GPU hours and model FLOPs to help reproducibility.

5.7 Human Evaluation

For generative QA and ambiguous episodic queries (e.g., multiple candidate episodes), perform human evaluations:

- Rate responses on correctness, helpfulness, and privacy concerns.
- Annotate failure modes (hallucination, incorrect timestamping, privacy leakage).

6 Context Window Augmentation via Distributed Mini-RAG Retrievals

A major limitation of current LLMs lies in their fixed context window size, typically on the order of a few thousand tokens, which restricts their ability to reason over long-term multimodal streams.

6.1 Mini-RAG Episodic Storage

We propose to segment incoming multimodal tokens into episodic memory units called *mini-RAGs* (Retrieval-Augmented Generators). Each mini-RAG acts as a lightweight local memory, storing tokens from a specific temporal or semantic segment (e.g., a morning session, a conversation, or a detected event).

Each mini-RAG maintains its own compressed representation and a small, efficient LLM for local reasoning and summarization. This structure enables scalable episodic storage that can extend indefinitely.

6.2 Hierarchical Querying and Retrieval

When a user query arrives at the central LLM, it first parses the query into tokens and performs an initial coarse retrieval step to identify relevant mini-RAGs by matching query tokens with indexed episodic tokens. This filtering reduces the search space drastically.

Subsequently, the central LLM queries the selected mini-RAGs, each of which performs local retrieval and reasoning, returning distilled relevant tokens or summaries.

6.3 Benefits and Scalability

This hierarchical retrieval architecture decouples long-term memory from the fixed LLM context window. It allows practically infinite context length, limited only by storage capacity and retrieval latency.

The use of a shared tokenizer ensures consistent token semantics across modalities and time, facilitating efficient matching and fusion during retrieval.

7 Vision Encoder: Tokenization and Temporal Modeling

The vision modality presents unique challenges due to high dimensionality, continuous input, and temporal dynamics.

7.1 Novel Tokenization Strategy

Instead of converting entire images into text descriptions, we propose a learnable tokenizer that converts image patches or video frames into discrete tokens aligned with the shared vocabulary. This tokenizer is trained jointly with the shared tokenizer to encode concepts such as objects, actions, and contextual cues.

These tokens capture semantic entities (e.g., ”car”, ”person”, ”moving object”) and their attributes (position, motion vector).

7.2 Temporal Modeling

To represent temporal continuity and motion, the vision encoder incorporates mechanisms for tracking tokens across frames. Tokens representing persistent objects are linked temporally, enabling the model to detect motion, occlusions, and changes.

We propose augmenting tokens with temporal embeddings and motion vectors, allowing the LLM to infer dynamic events and avoid redundant token generation for static scenes.

7.3 Applications

This tokenization enables efficient streaming of live video into the shared token space, supporting long-term scene understanding, action recognition, and multimodal fusion with audio and text tokens.

8 Audio Encoder: Semantic Tokenization and Event Detection

The audio modality complements vision by providing rich temporal and contextual cues, such as speech, environmental sounds, and events invisible to vision.

8.1 Semantic Audio Tokenization

Raw audio waveforms are first transformed into spectrograms or Mel-frequency cepstral coefficients (MFCCs). A learnable audio tokenizer then converts these features into discrete tokens aligned with the shared vocabulary. Tokens represent distinct sound events (e.g., ”engine noise”, ”footsteps”, ”speech”) or acoustic scenes.

Joint training with the shared tokenizer ensures that semantically similar audio and vision events produce closely aligned token embeddings, facilitating cross-modal fusion.

8.2 Event Detection and Temporal Segmentation

The audio encoder incorporates temporal convolutional or transformer layers to detect event boundaries and segment continuous streams into meaningful chunks. Tokens are augmented with temporal embeddings encoding event onset, duration, and confidence.

This structured token stream enables downstream modules to perform precise cross-modal reasoning about events co-occurring in time.

9 Central LLM: Cross-Modal Attention and Dynamic Input Fusion

The core of the *SENSE* architecture is the central Large Language Model (LLM) responsible for integrating multimodal information and performing complex reasoning.

9.1 Cross-Modal Attention Mechanisms

Unlike traditional LLMs processing text alone, our central LLM is designed to attend jointly over tokens originating from multiple modalities — vision, audio, and text — all embedded in a shared token space.

We adopt a cross-modal attention mechanism that assigns dynamic weights to tokens depending on their relevance to the current query and context. Formally, given queries Q, keys K, and values V concatenated from all modalities, the attention is computed as:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right) V$$

where M is a learned modality bias matrix encoding cross-modal correlations, and d_k is the key dimension.

This formulation allows the model to focus on complementary or reinforcing signals, e.g., linking the visual token for ”car” with the audio token for ”engine noise”.

9.2 Dynamic Input Gating

To manage the large volume of tokens from multiple modalities, the model incorporates input gating layers that learn to suppress irrelevant or redundant tokens dynamically. This enhances efficiency and reduces noise.

Gating weights are learned end-to-end during training, conditioned on the query and historical context.

9.3 Multi-Head and Hierarchical Attention

Building on the Transformer backbone, multi-head attention heads specialize in attending to different modalities or temporal segments. A hierarchical attention mechanism further aggregates information at various granularities — token-level, segment-level (mini-RAG), and global context — facilitating scalable reasoning.

9.4 Training Regime and Objectives

The central LLM is trained with a combination of:

- **Masked Token Prediction** over the shared token stream to learn coherent multimodal representations.
- **Cross-Modal Alignment Losses** enforcing semantic consistency between modalities.
- **Retrieval-Augmented Generation** objectives where the model learns to query mini-RAGs effectively.

Training utilizes large-scale multimodal datasets with synchronized video, audio, and textual annotations (e.g., YouTube-8M, AVSD).

10 Textual LLM: Foundation and Adaptation

The textual LLM forms the language backbone, pre-trained on vast corpora and adapted for multimodal grounding.

10.1 Foundation Model

We leverage state-of-the-art transformer-based LLM architectures (e.g., GPT, LLaMA) pre-trained with causal or masked language modeling objectives on large-scale text datasets.

10.2 Multimodal Adaptation

To align the textual LLM with vision and audio modalities, fine-tuning is performed on datasets providing synchronized multimodal tokens mapped through the shared tokenizer.

Adapters and cross-attention layers are introduced to process and integrate non-textual token streams while preserving core language modeling capabilities.

10.3 Training Objectives

The textual LLM training objectives include:

- **Masked Language Modeling (MLM):** Predict missing tokens conditioned on multimodal context.
- **Next Token Prediction:** Maintain autoregressive generation fluency.
- **Multimodal Consistency Loss:** Enforce semantic alignment between text and non-text tokens.

10.4 Inference and Generation

At inference, the textual LLM conditions on the fused token representations provided by the central LLM and mini-RAG queries, enabling grounded and context-aware natural language generation.

11 Training Data and Scaling Strategies

11.1 Multimodal Dataset Requirements

The *SENSE* model requires large-scale datasets containing temporally aligned video, audio, and text streams. Public datasets such as YouTube-8M, AVSD, and HowTo100M provide billions of annotated video/audio segments paired with natural language captions or transcripts.

These datasets offer diverse scenarios, environments, and activities, which is crucial for learning generalized cross-modal representations.

11.2 Data Preprocessing and Tokenization

All modalities are tokenized using the shared tokenizer, transforming raw inputs into discrete token sequences that preserve semantic information while enabling joint modeling.

Video frames and audio clips are preprocessed through modality-specific encoders to generate embeddings compatible with the shared token vocabulary.

11.3 Scaling Considerations

Training *SENSE* at scale involves:

- **Distributed Training**: Leveraging large GPU clusters with data and model parallelism to handle the immense token volume.
- **Curriculum Learning**: Starting from unimodal pretraining, progressively incorporating multimodal data to stabilize convergence.
- **Memory Efficiency**: Using techniques such as mixed precision training, gradient checkpointing, and sparse attention to manage GPU memory.
- **Fine-Tuning**: Continual adaptation on domain-specific data to refine performance.

11.4 Privacy and Security

Given the extensive use of real-world data, privacy-preserving techniques such as federated learning and differential privacy should be investigated to mitigate risks of sensitive information leakage.

12 Discussion

12.1 Limitations and Challenges

While promising, *SENSE* faces several challenges:

- **Computational Cost**: Training and inference over massive multimodal token streams require extensive resources.
- **Synchronization Precision**: Aligning modalities at scale can introduce noise; slight temporal misalignments impact performance.
- **Catastrophic Forgetting**: Continuous updating of mini-RAG memories risks overwriting relevant information without careful management.
- **Data Biases**: Public datasets contain inherent cultural and contextual biases that affect generalization.

12.2 Privacy and Ethical Considerations

The extensive use of real-world audiovisual data necessitates stringent privacy controls and ethical review, especially if deployed in sensitive contexts.

12.3 Broader Impact

By unifying multiple sensory streams into a shared representational space, *SENSE* paves the way toward AI systems with richer situational awareness and human-like perception capabilities, which could revolutionize applications in robotics, assistive technology, and immersive media.

13 Conclusion and Future Work

This paper introduces *SENSE*, a novel architecture unifying vision, audio, and language processing through shared tokenization and distributed episodic memory (mini-RAGs), fused by a central LLM employing hierarchical cross-modal attention.

We demonstrate the conceptual foundations, architectural design, and training paradigms necessary to realize a truly multimodal AI system with extended memory capabilities.

Future directions include:

- Extending to additional modalities (e.g., tactile, proprioception).
- Optimizing mini-RAG storage for lifelong continual learning.
- Exploring privacy-preserving training at scale.
- Developing more efficient hierarchical attention mechanisms.
- Conducting comprehensive real-world deployment studies.

We invite the community to build upon *SENSE*'s framework to advance towards more holistic and context-aware AI.

References

[1] Radford, A., et al. Learning transferable visual models from natural language supervision. *International conference on machine learning*, 2021.

[2] Girdhar, R., et al. ImageBind: One embedding space to bind them all. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[3] Huang, S., et al. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*, 2023.

[4] Alayrac, J.B., et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 2022.

[5] Reed, S., et al. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[6] Lewis, P., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 2020.

[7] Zhu, D., et al. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[8] Blum, et al. EpiNet: Episodic memory for continual learning. *Conference paper*, 2022.