Data Ingestion Plan & Chunking Strategy

Overview of the Commercial Payload Challenge

- **Metadata Complexity:**

  The core payload comprises 18 fields based on DDEX/CWR standards. Fixed fields (e.g., ISRC, ISWC, Duration) are relatively compact (~482 bits total), but variable fields (e.g., Title, Performer, Ownership Information) can balloon the total size to a worst-case estimate of roughly 27,792 bits (or ~3,474 bytes). In extreme cases, payloads may exceed even one-second encoding capacity.

- **Optimization Imperative:**

  To handle such dynamic and potentially massive payloads, we must:

  - Use dynamic bit allocation (e.g., Huffman or Lempel-Ziv encoding) for variable-length fields.
  - Split the payload into core versus auxiliary metadata.
  - Leverage hierarchical structures that embed higher-level IDs or pointers to external resources.
  - Adjust model training and architecture to account for sequential bit processing and variable-length data.

---

1. Data Ingestion Plan

Objective:
Establish a robust ingestion pipeline that extracts, validates, and normalizes metadata while preparing it for dynamic encoding and ensuring audio-metadata synchronization.

A. Ingestion Pipeline Design

1. **Source Integration & Extraction:**
   - **Multi-Source Integration:**

     Connect to upstream feeds and databases that supply the 18 metadata fields. Ensure support for both fixed fields (ISRC, ISWC, Duration, etc.) and variable fields (Title, Performer(s), Publisher(s), etc.).

   - **Modular Architecture:**

     Create distinct modules for extraction, transformation, and validation to isolate issues and enable iterative refinement.

2. **Validation & Quality Assurance:**
    - **Schema Enforcement:**

      Define and enforce a strict schema (including field character limits and required formats). Automated validators should flag missing or malformed fields.

    - **Edge Case Handling:**

        - For variable-length fields, check that character counts do not exceed limits (up to 255 characters).
        - Log any deviations so that manual intervention or automated imputation can be applied.

    - **Automated Testing:**

      Include unit and integration tests simulating worst-case payloads to ensure the pipeline can handle high variability.

3. **Normalization & Efficient Encoding:**
    - **Dynamic Bit Allocation:**

      Apply compression techniques (e.g., Huffman or Lempel-Ziv) to variable-length fields. This reduces the effective bit usage by encoding common characters with shorter bit patterns.

    - **Efficient Representations:**

      Explore coding schemes or indexing for fields with multiple entries (like multiple performers or rights holders), so that instead of raw text, we can use reference codes or indexes to external lookups.

    - **Preprocessing:**

      Standardize encodings and clean the data to ensure compatibility with downstream encoding modules.

4. **Audio-Metadata Synchronization:**
    - **Sync Marker Insertion:**

      Embed synchronization markers into each 1-second audio segment during ingestion. These markers will later be used to align micro-chunks for decoding.

    - **Tight Coupling:**

      Ensure that each segment's metadata payload is directly linked to its corresponding audio, so even if segments are trimmed or compressed, the payload remains extractable.

5. **Monitoring & Feedback:**
    - **Dashboard & Logging:**

      Implement real-time monitoring to track ingestion success, validation errors, and payload size metrics.

    - **Iterative Improvement:**

      Feed back results from downstream encoding/decoding tests to continuously refine preprocessing and normalization rules.

---

2. Chunking Strategy

Objective:
Develop a resilient method to split the complete payload (including potentially large variable-length fields) into micro-chunks that are redundantly embedded across each 1-second segment of audio, ensuring robust recovery even under adverse conditions.

A. Payload Splitting & Encoding

1. **Core vs. Auxiliary Metadata Partitioning:**
    - **Core Metadata:**

      Includes critical fields (ISRC, Title, Duration) that must be recoverable with the highest reliability. These are prioritized and allocated a fixed, robust encoding (approximately ~500 bytes total).

    - **Auxiliary Metadata:**

      Covers extended fields (Ownership Information, License Terms, Artist Roles, etc.), which can be compressed and distributed using dynamic bit allocation. This portion may use hierarchical encoding to include pointers or indexes for external lookups.

2. **Fixed Micro-Chunks with Overlap:**
    - **Chunk Formation:**

      Serialize and compress the full payload and split it into fixed micro-chunks of about 10–20 ms each.

    - **Overlapping Distribution:**

      For example, within a 1-second segment:

        - 0.0–0.5 seconds: Chunks A, B, C, D
        - 0.5–1.0 seconds: Chunks C, D, E, F

      This overlapping strategy ensures redundancy and error resilience.

3. **Time-Based & Frequency-Based Encoding:**

    - **Time-Based Redundancy:**

      Each 1-second segment is designed to be self-contained, holding all necessary micro-chunks to reconstruct the full payload.

    - **Frequency-Based Allocation:**

        - **Core Fields:** Embedded in robust, low/mid frequency bands.
        - **Auxiliary Fields:** Distributed across higher frequencies with additional error correction.

    - **Error Correction:**

      Use techniques such as Reed-Solomon coding to mitigate bit errors resulting from compression, clipping, or other distortions.

B. Advanced Optimization & Model Architecture

1. **Dynamic Chunking & Hybrid Approaches:**

    - **Adaptive Chunking:**

      For payloads exceeding the typical ~3,000-byte size (as in worst-case scenarios), consider a hybrid strategy:

        - Use fixed micro-chunks for core metadata.
        - Implement dynamic or sequential chunking for auxiliary data that can be spread across multiple segments or referenced via hierarchical pointers.

    - **Hierarchical Metadata Management:**

      Embed higher-level IDs or indices that point to extended metadata stored externally. This reduces the immediate payload size while preserving access to full information.

2. **Model Architecture Adjustments:**

    - **Invertible Networks:**

      Ensure that the network architecture can handle full payloads by enabling dynamic bit allocations and sequential processing.

    - **Side Information:**
      Incorporate additional side information in the network to decode complex metadata structures, improving overall accuracy.

3. **Training and Curriculum Adjustments:**

    - **Stage 1 – Core Functionality:**
      Begin training with only the core metadata to achieve low distortion and reliable invertibility. This stage may require an additional 5,000–10,000 steps.
    - **Stage 2 – Variable-Length Payloads:**
      Gradually introduce variable-length and auxiliary fields, allowing the network to adapt. This may take an additional 10,000–20,000 steps.
    - **Stage 3 – Representation Learning:**
      Fine-tune the model to learn optimal representations for different types of metadata (numeric vs. textual) with another 20,000–30,000 steps.

    - **Loss Function Tuning:**

        - Implement weighted loss functions to prioritize recovery of essential fields (e.g., track ID, ISRC).
        - Fine-tune loss functions to minimize bit error rates for all critical fields.
    - **Overall Training Increase:**

      Expect an additional ~60,000 training steps on top of the baseline (~1.26 million steps) to incorporate these optimizations.

Summary

Data Ingestion Plan:

- Build a modular, validated ingestion pipeline that extracts and normalizes both fixed and variable metadata fields.
- Apply dynamic bit allocation and compression to variable-length fields.
- Insert synchronization markers in each 1-second audio segment to tightly couple metadata with audio.

Chunking Strategy:

- Partition the payload into core (fixed) and auxiliary (variable) components.
- Split the full payload into overlapping fixed micro-chunks (10–20 ms each) that are redundantly embedded in every 1-second segment.
- Use time-based and frequency-based encoding, along with robust error correction, to ensure complete recovery.
- Explore hybrid chunking approaches and adjust model architecture and training (with a curriculum learning approach) to accommodate variable payload sizes and optimize bit usage.