

Opus

A Workflow Intention Framework for Complex Workflow Generation

Phillip Kingston¹
Member of Technical Staff
AppliedAI

Théo Fagnoni¹
Member of Technical Staff
AppliedAI

Mahsun Altin²
Member of Technical Staff
AppliedAI

25 January 2025



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0)

Abstract

This paper introduces Workflow Intention, a novel framework for identifying and encoding process objectives within complex business environments. Workflow Intention is the alignment of Input, Process and Output elements defining a Workflow’s transformation objective interpreted from Workflow Signal inside Business Artefacts. It specifies how Input is processed to achieve desired Output, incorporating quality standards, business rules, compliance requirements and constraints. We adopt an end-to-end Business Artefact Encoder and Workflow Signal interpretation methodology involving four steps: Modality-Specific Encoding, Intra-Modality Attention, Inter-Modality Fusion Attention then Intention Decoding. We provide training procedures and critical loss function definitions. In this paper:

1. We introduce the concepts of Workflow Signal and Workflow Intention, where Workflow Signal—decomposed into Input, Process and Output elements—is interpreted from Business Artefacts, and Workflow Intention is a complete triple of these elements.
2. We introduce a mathematical framework for representing Workflow Signal as a vector and Workflow Intention as a tensor, formalizing properties of these objects.
3. We propose a modular, scalable, trainable, attention-based multimodal generative system to resolve Workflow Intention from Business Artefacts.

¹ Equal contribution, corresponding author: phillip.kingston@opus.com

² Minor contribution

1 Introduction

In the lifecycle of most businesses and government organizations, developing and refining internal processes is crucial for maintaining structure, consistency, and overall operational efficiency. Well-defined processes offer tangible advantages: they enhance quality control, reduce costs, mitigate risks, and facilitate auditing. They also help ensure continuity when employees retire or transition out of the organization. Furthermore, many regulatory and compliance agencies publish procedural guidelines to help external stakeholders understand and meet specific standards.

Despite the clear benefits of process documentation, its quality, completeness, currency, accuracy, and granularity can vary widely across organizations. This is where the concept of Workflow Intention becomes invaluable. By extracting the core purpose and objectives from existing Business Artefacts—such as standard operating procedures, policy manuals, and historical records—Workflow Intention enables organizations to rapidly implement supervised automation and evolve legacy processes into efficient AI-enhanced Workflows enriched with best practice.

Definitions Our methodology is based on the following concepts introduced in *Opus: A Large Workflow Model for Complex Workflow Generation* by Fagnoni et al. [1]:

Input: The dataset initiating a **Process**, conforming to validation rules and format specifications. **Input** is multimodal, including structured (e.g. databases, forms) and unstructured (e.g. documents, media) data types such as text, documents, images, audio, and video.

Process: A structured sequence of operational steps transforming **Input** into **Output**, defined in part or whole across **Business Artefacts**. **Process** combines automated and manual steps defining start/end conditions, decision points, parallel paths, roles, success criteria, monitoring, metrics, compliance requirements and error handling.

Output: The result of a **Process** operating on **Input**, meeting predefined quality and business criteria. **Output** can be tangible (e.g. documents) or intangible (e.g. decisions) and include audit trails of their creation. Supported formats include text and documents.

Business Artefact: Any text, document, image, audio, or video capturing **Process** knowledge (e.g. process maps, standard operating procedures, regulatory guidelines, compliance documents) or that provides **Workflow Signals**. **Business Artefacts** detail input/output specifications, task sequences, business rules and roles.

Workflow: A software defined executable **Process** as a sequence of **Tasks**. **Workflows** coordinate task execution, manage data flow, and enforce business rules, compliance, and process logic (e.g. conditionals, loops, error handling). **Workflows** support monitoring, logging, audit, state management, concurrency, adaptive modification, and version control.

Task: An atomic unit of work within a **Workflow**, performing a specific function with defined input/output schemas, objectives, timing constraints, and success criteria. **Tasks** adhere to the singular responsibility principle, support automation or manual intervention, and maintain contextual awareness of dependencies. **Tasks** are auditable by humans or AI agents against their definition.

Workflow Intention: The alignment of **Input**, **Output**, and **Process** components defining a **Workflow**’s transformation objective. It specifies how **Input** is processed to achieve desired **Output**, incorporating data formats, quality standards, business rules, and constraints. It is determined by interpreting **Workflow Signals** from direct and indirect sources.

Workflow Signal: A discrete informational cue from **Business Artefacts** or **Intention Elicitation** that conveys implicit or explicit information on **Input**, **Process** or **Output** relevant to a **Workflow**.

Intention Elicitation: User-driven communication (e.g. text-based conversations, interface interactions) that contains **Workflow Signals** to further articulate **Workflow Intention(s)**. It captures objectives, constraints, and preferences, distinct from **Business Artefacts** and **Input/Output** examples.

Complete Intention: The state where sufficient information exists across **Input**, **Output**, and **Process** components for accurate **Workflow** implementation. Incomplete intentions lack clear specifications, relationships, or operational requirements, hindering execution.

Mixed Intention: The state where **Business Artefacts** or **Intention Elicitation** describe multiple distinct transformation objectives, requiring separation into individual **Workflow Intentions**. Separation improves clarity, maintainability, and preserves **Workflow** interfaces.

This paper makes the following key contributions:

1. We introduce the concepts of Workflow Signal and Workflow Intention. Workflow Signals are interpreted from Business Artefacts and decomposed into Input (i), Process (p) and Output (o) elements. Workflow Intention is defined as a triple of Workflow Signals i , p and o .
2. We introduce a mathematical framework for representing Workflow Signals as vectors i, p, o and Workflow Intention as a tensor (i, p, o) , formalizing properties of these objects under this framework.
3. We propose a modular, scalable, trainable, attention-based multimodal generative system to resolve Workflow Intention.

2 Background

This framework leverages many state-of-the-art methodologies to obtain contextual representations of multimodal Business Artefacts in order to generate Workflow Intention. Transformer-based architectures introduced a paradigm shift from recurrence to parallelizable self-attention, enabling efficient long-range dependency modeling and significantly improving scalability in natural language processing. Subsequent innovations in encoder-only (RoBERTa), decoder-only (GPT), and encoder-decoder (T5) architectures further refined contextual representation and generation capabilities. Advancements in visual (ViT) and multimodal (FLAVA, NVLM, InternVL) transformers have broadened these capabilities to handle image and text jointly, providing robust backbones for extracting structured features from Business Artefacts to facilitate Workflow Intention generation.

Transformer Architecture The transformer model introduced by Vaswani et al. [2] replaced recurrence with self-attention, allowing tokens to attend globally within a sequence. This enabled parallel processing and improved long-range dependency modeling, with the multi-head attention mechanism capturing diverse relationships. By eliminating step-by-step processing, transformers became highly scalable and efficient for NLP tasks. This mechanism is central to our approach as we aim to extract dense features from long-range sequences produced by Business Artefacts.

Encoder, Decoder, Encoder-Decoder RoBERTa by Liu et al. [3] is an optimized encoder-only architecture designed to enhance contextual representations in language models. It refines pretraining strategies by removing next sentence prediction, incorporating dynamic masking and extending training on larger datasets. As an encoder-based model, it effectively captures rich representations, which we leverage to preserve features consistently across our Workflow Intention generation pipeline.

Decoder-only architectures, such as GPT, rely exclusively on a causal decoder to generate text recursively. Unlike encoder models, GPT processes input unidirectionally, meaning tokens attend only to previous tokens, ensuring autoregressive generation. The decoder generates text one token at a time, using a stopping mechanism to determine completion. We employ a decoder to generate vectors based on our framework to represent Workflow Intention.

T5 by Raffel et al. [4] extends transformer capabilities by combining an encoder and decoder with cross-attention, where the decoder attends to encoded representations before generating output. Unlike GPT, which generates step by step based on past tokens, T5 benefits from a bidirectional encoder, capturing full context before passing information to the decoder. This architecture allows us to generate Workflow Intention based on a context captured from Business Artefacts.

Visual Transformers The vision transformer adapted transformers for images by dividing input images into fixed-size patches, treating them as tokens, and applying self-attention. This allowed the model to capture both local and global relationships without convolutions. ViT by Dosovitskiy et al. [5] demonstrated that a pure attention-based approach can match or surpass CNNs on vision tasks when trained on large datasets, proving the generalizability of transformers beyond NLP. We leverage these architectures to process document and image Business Artefacts.

Multimodal Transformers Multimodal transformers integrate text and vision, allowing AI models to reason across modalities. FLAVA by Singh et al. [6] combines separate image and text encoders with a multimodal encoder to align representations for captioning and visual question-answering tasks. T5-inspired architectures for vision-language tasks leverage cross-attention to fuse textual and visual embeddings effectively. NVLM by Dai et al. [7], a large multimodal LLM, integrates vision encoders into an LLM while maintaining strong language capabilities, excelling at tasks requiring both modalities. InternVL by Chen et al. [8] scales multimodal learning further, progressively aligning large vision models with text models to handle diverse inputs, including video and complex multimodal reasoning. These state-of-the-art scalable architectures serve as backbones for ingesting multimodal Business Artefacts, constructing context from them, and generating Workflow Intention.

3 System Overview

Let $\mathcal{M} = \{m_1, m_2, \dots, m_K\}$ denote the set of K distinct modalities. For each modality m_k , we consider a set of Business Artefacts $\mathcal{A}_{m_k} = \{a_{m_k,1}, a_{m_k,2}, \dots, a_{m_k,N_{m_k}}\}$, where N_{m_k} is the number of Business Artefacts in modality m_k . Each Business Artefact $a_{m_k,i}$ is represented in its raw form, with modality-specific dimensionality. In this paper, we consider three modalities: Text (T), Image (I) and Document (D). The framework is built to support any modality.

Each Business Artefact from \mathcal{A}_{m_k} gets encoded in a dedicated pipeline. Then, all encoded Business Artefacts from \mathcal{A}_{m_k} are concatenated and encoded by a dedicated intra-modality pipeline. The encoded Business Artefacts are then concatenated across the modalities by a fusion encoder, before entering the Workflow Intention decoder. The encoded decoder generates vectors which are projected into Workflow Intention objects until it stops.

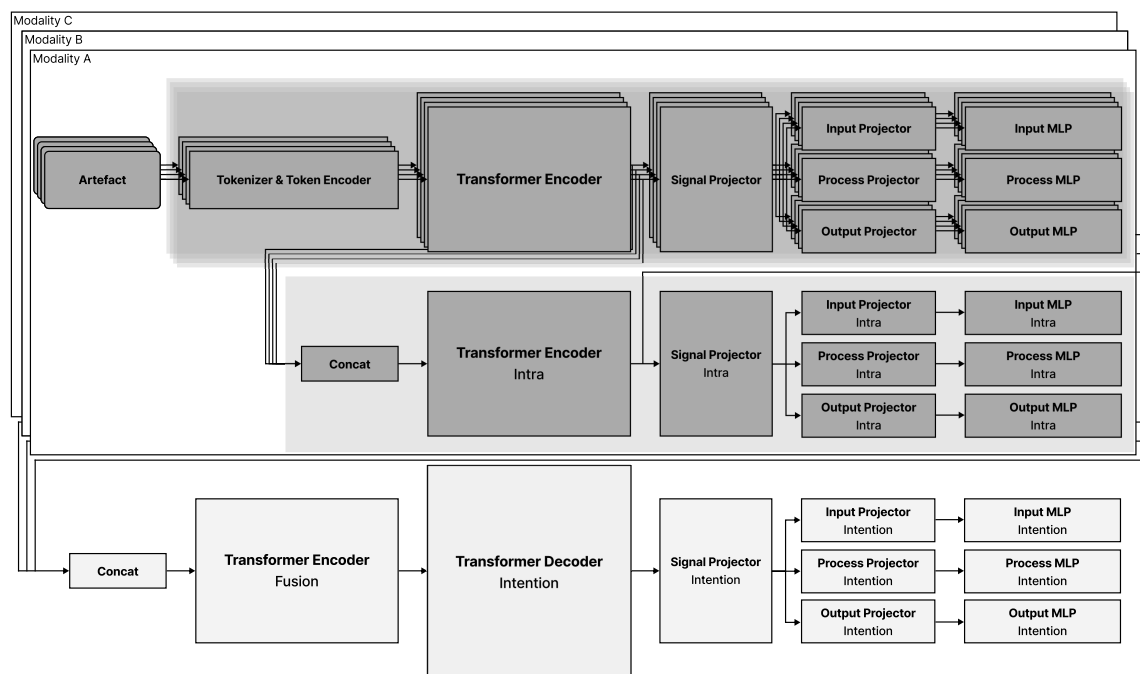


Figure 1: High-Level System Overview

4 Business Artefacts Encoding and Signal Extraction

Each Business Artefact $a_{m_k,i}$ of modality m_k is tokenized by a tokenizer \mathbf{Tok}_{m_k} . Each token is embedded by a token encoder \mathbf{E}_{m_k} . The resulting sequence of vectors $\mathbf{E}_{m_k,i}$ is encoded by a self-attention based encoder $\mathbf{Encoder}_{m_k}$ into $\mathbf{E}_{m_k,i}^{\text{enc}}$. The encoded sequence is projected into a unified space via a linear projection parametrized by $\mathbf{W}_{m_k}^u, b_{m_k}^u$ and the arrival dimension d , giving $\mathbf{H}_{m_k,i}$. A learned representative vector $h_{i,m_k}^{[\text{REP}]}$ is retrieved or computed from $\mathbf{H}_{m_k,i}$ and linearly projected by three Projection Heads: Input ($\mathbf{W}_{m_k}^I, b_{m_k}^I$), Process ($\mathbf{W}_{m_k}^P, b_{m_k}^P$) and Output ($\mathbf{W}_{m_k}^O, b_{m_k}^O$).

4.1 Modality-Specific Business Artefact Encoding

4.1.1 Text Business Artefacts

Let T be the text modality. For ease of notation, let $T_i = a_{T,i}$ a text Business Artefact.

Text Tokenizer and Token Encoder We tokenize T_i with \mathbf{Tok}_T into a sequence of L_{T_i} tokens $T_i \mapsto \{[\text{CLS}]_T, t_{i,1}, t_{i,2}, \dots, t_{i,L_{T_i}-1}\}$, prepending a $[\text{CLS}]_T$ token. Each token $t_{i,j}$ is mapped to an embedding vector $\mathbf{E}_T(t_{i,j}) \in \mathbb{R}^{d_T}$ by a text token encoder $\mathbf{E}_T : \mathcal{V}_T \rightarrow \mathbb{R}^{d_T}$ where \mathcal{V}_T is the vocabulary (the set of all possible text tokens) and d_T is the embedding dimensionality. We define the tensor representation $\mathbf{E}_{T_i} = \mathbf{E}_{T,i}$ for ease of notation as the sequence of embedded tokens:

$$\mathbf{E}_{T_i} = [\mathbf{E}_T([\text{CLS}]_T), \mathbf{E}_T(t_{i,1}), \mathbf{E}_T(t_{i,2}), \dots, \mathbf{E}_T(t_{i,L_{T_i}-1})] \in \mathbb{R}^{d_T \times L_{T_i}} \quad (1)$$

Text Encoder To capture contextual dependencies across all tokens in T_i , we pass \mathbf{E}_{T_i} through an encoder network $\mathbf{Encoder}_T$ to calculate $\mathbf{E}_{T_i}^{\text{enc}} = \mathbf{E}_{T,i}^{\text{enc}}$ in which each column contains a contextualized embedding of the corresponding token. Finally, we employ a linear projection to map $\mathbf{E}_{T_i}^{\text{enc}}$ into a d -dimensional space. The projection is achieved by applying a learnable weight matrix $\mathbf{W}_T^u \in \mathbb{R}^{d \times d_T}$ and a bias vector $b_T^u \in \mathbb{R}^d$, thus forming the final representation \mathbf{H}_{T_i} :

$$\mathbf{H}_{T_i} = \mathbf{W}_T^u \mathbf{E}_{T_i}^{\text{enc}} + b_T^u, \quad \mathbf{H}_{T_i} \in \mathbb{R}^{d \times L_{T_i}} \quad (2)$$

4.1.2 Image Business Artefacts

Let F be the image modality. For ease of notation, let $F_i = a_{F,i}$ an image Business Artefact. F_i is represented in its raw form as a three-dimensional tensor of size $\mathbb{R}^{c \times h \times w}$, where c denotes the number of channels, h and w represent the height and width of the image, respectively.

Image Tokenizer and Token Encoder To remain consistent with the tokenizing abstraction, i.e. “tokenizer” for text, we adopt the same terminology, even though it is actually a specific image feature extraction described in the appendix.

We tokenize F_i with \mathbf{Tok}_F into a sequence of L_{F_i} patches $F_i \mapsto \{f_{i,1}, f_{i,2}, \dots, f_{i,L_{F_i}}\}$. We do not employ any $[\text{CLS}]$ -type vector representation, consistent with ViT (Dosovitskiy et al. [5]). Each patch $f_{i,j}$ is mapped to an embedding vector $\mathbf{E}_F(f_{i,j}) \in \mathbb{R}^{d_F}$ by an image patch encoder $\mathbf{E}_F : \mathbb{R}^{c \times h \times w} \rightarrow \mathbb{R}^{d_F}$ where d_F is the embedding dimensionality.

We define the tensor representation $E_{F_i} = E_{F,i}$ for ease of notation as the sequence of embedded patches:

$$E_{F_i} = [\mathbf{E}_F(f_{i,1}), \mathbf{E}_F(f_{i,2}), \dots, \mathbf{E}_F(f_{i,L_{F_i}})] \in \mathbb{R}^{d_F \times L_{F_i}} \quad (3)$$

Image Encoder To capture contextual dependencies across all patches in F_i , we pass E_{F_i} through an encoder network $\mathbf{Encoder}_F$ to calculate $E_{F_i}^{\text{enc}} = E_{F,i}^{\text{enc}}$ in which each column contains a contextualized embedding of the corresponding token. Finally, we employ a linear projection to map $E_{F_i}^{\text{enc}}$ into a d -dimensional space. The projection is achieved by applying a learnable weight matrix $W_F^u \in \mathbb{R}^{d \times d_F}$ and a bias vector $b_F^u \in \mathbb{R}^d$, thus forming the final representation H_{F_i} :

$$H_{F_i} = W_F^u E_{F_i}^{\text{enc}} + b_F^u, \quad H_{F_i} \in \mathbb{R}^{d \times L_{F_i}} \quad (4)$$

4.1.3 Document Business Artefacts

Let D be the document modality. We treat each document page as an image of size (h_D, w_D) . Each page is tokenized separately. For ease of notation, let $D_{p,i} = a_{D,p,i}$ a page of a document Business Artefact and $D_i = a_{D,i}$ a document Business Artefact.

Document Page Tokenizers and Token Encoders $D_{p,i}$ is decomposed as (inspired by the method of NVLM by Dai et al. [7]):

- Text elements $\{D_{p,i,q}^T\}_q$: a set of text elements, each tokenized through \mathbf{Tok}_T then concatenated, prepended with a $[\text{CLS}]_T$ token, producing a sequence of $L_{D_{p,i}}^T$ tokens $\{d_{i,j}^T\}_j$. Each token $d_{i,j}^T$ is mapped to an embedding vector $\mathbf{E}_T(d_{i,j}^T) \in \mathbb{R}^{d_T}$ using $\mathbf{E}_T: \mathcal{V}_T \rightarrow \mathbb{R}^{d_T}$, producing a sequence of vectors denoted by $E_{D_{p,i}}^T \in \mathbb{R}^{d_T \times L_{D_{p,i}}^T}$.
- Text spatial elements: bounding box coordinates of each text token $d_{i,j}^T$ expressed in text as $\text{box}_{i,j}^T = \langle \langle \text{box} \rangle (x_{\min,i,j}^T, y_{\min,i,j}^T), (x_{\max,i,j}^T, y_{\max,i,j}^T) \rangle \langle / \text{box} \rangle$, with $\text{box}_{i,[\text{CLS}]_T}^T = \langle \langle \text{box} \rangle ((0, 0), (h_D, w_D)) \rangle \langle / \text{box} \rangle$, and mapped to an embedding $\mathbf{E}_S(\text{box}_{i,j}^T) \in \mathbb{R}^{d_T}$, producing a sequence of vectors denoted by $E_{D_{p,i}}^{T_s} \in \mathbb{R}^{d_T \times L_{D_{p,i}}^T}$. $\mathbf{E}_S = \overline{\mathbf{Encoder}_T}$ denotes the average over the sequence of embeddings produced by the text encoder, in order to obtain one embedding per bounding box.
- Image elements $\{D_{i,q}^F\}_q$: a set of image elements, each patched through \mathbf{Tok}_F then concatenated, producing a sequence of $L_{D_{p,i}}^F$ patches $\{d_{i,j}^F\}_j$. Each patch $d_{i,j}^F$ is mapped to an embedding vector $\mathbf{E}_F(d_{i,j}^F) \in \mathbb{R}^{d_F}$ using $\mathbf{E}_F: \mathbb{R}^{c \times h \times w} \rightarrow \mathbb{R}^{d_F}$, producing a sequence of vectors denoted by $E_{D_{p,i}}^F \in \mathbb{R}^{d_F \times L_{D_{p,i}}^F}$.
- Image spatial elements: bounding box coordinates of each image patch $d_{i,j}^F$ expressed in text as $\text{box}_{i,j}^F = \langle \langle \text{box} \rangle (x_{\min,i,j}^F, y_{\min,i,j}^F), (x_{\max,i,j}^F, y_{\max,i,j}^F) \rangle \langle / \text{box} \rangle$, with $\text{box}_{i,[\text{CLS}]_F}^F = \langle \langle \text{box} \rangle (0, 0), (h_D, w_D) \rangle \langle / \text{box} \rangle$, and mapped to an embedding $\mathbf{E}_S(\text{box}_{i,j}^F) \in \mathbb{R}^{d_T}$, producing a sequence of vectors denoted by $E_{D_{p,i}}^{F_s} \in \mathbb{R}^{d_T \times L_{D_{p,i}}^F}$. $\mathbf{E}_S = \overline{\mathbf{Encoder}_T}$ denotes the average over the sequence of embeddings produced by the text encoder, in order to obtain one embedding per bounding box.

Since the dimensions d_T and d_F may differ, we project each patch embedding into \mathbb{R}^{d_T} via a learnable linear projection with bias term ($W_F^D \in \mathbb{R}^{d_T \times d_F}$, $b_F^D \in \mathbb{R}^{d_T}$) to obtain

$$\tilde{E}_{D_{p,i}}^F = W_F^D E_{D_{p,i}}^F + b_F^D \in \mathbb{R}^{d_T \times L_{D_{p,i}}^F} \quad (5)$$

Document text and image element embeddings are concatenated with their respective spacial element embeddings to produce a sequence of vectors as follows:

$$E_{D_{p,i}} = \text{Concat}(E_{D_{p,i}}^{\text{F, Concat}}, E_{D_{p,i}}^{\text{T, Concat}}) \in \mathbb{R}^{d_T \times L_{D_{p,i}}} \text{ with } L_{D_{p,i}} = 2(L_{D_{p,i}}^F + L_{D_{p,i}}^T) \quad (6)$$

where

$$E_{D_{p,i}}^{\text{F, Concat}} = \text{Concat}((W_F^D E_F(d_{i,j}^F) + b_F^D, \mathbf{E}_S(\text{box}_{i,j}^F)))_j \in \mathbb{R}^{d_T \times 2L_{D_{p,i}}^F} \quad (7)$$

$$E_{D_{p,i}}^{\text{T, Concat}} = \text{Concat}((E_T(d_{i,j}^T), \mathbf{E}_S(\text{box}_{i,j}^T)))_j \in \mathbb{R}^{d_T \times 2L_{D_{p,i}}^T} \quad (8)$$

Document Encoder The tokenized pages $\{E_{D_{p,i,n}}\}_n$ are concatenated into E_{D_i} . To capture contextual dependencies across all tokens and patches in D_i , we pass E_{D_i} through an encoder network **Encoder_D** to calculate $E_{D_i}^{\text{enc}} = E_{D_i}^{\text{enc}}$ in which each column contains a contextualized embedding of the corresponding token, token spatial coordinates, patch and patch spatial coordinates. Finally, we employ a linear projection to map $E_{D_i}^{\text{enc}}$ into a d -dimensional space. The projection is achieved by applying a learnable weight matrix $W_D^u \in \mathbb{R}^{d \times d_D}$ and a bias vector $b_D^u \in \mathbb{R}^d$, thus forming the final representation H_{D_i} :

$$H_{D_i} = W_D^u E_{D_i}^{\text{enc}} + b_D^u, \quad H_{D_i} \in \mathbb{R}^{d \times L_{D_i}} \quad (9)$$

4.2 Input, Process, Output Projection Heads

4.2.1 Text Artefact Originated Workflow Signals

The encoded [CLS] token representation of T_i , $h_{i, [\text{CLS}]_T} \in H_{T_i}$ is retrieved as the learned representative vector, $h_{i, T}^{[\text{REP}]} = h_{i, [\text{CLS}]_T}$ and linearly projected by three separate Projection Heads: Input (W_T^I, b_T^I), Process (W_T^P, b_T^P) and Output (W_T^O, b_T^O), to obtain the following Workflow Signals:

$$i_{T_i} = W_T^I h_{i, T}^{[\text{REP}]} + b_T^I \in \mathbb{R}^d \quad (10)$$

$$p_{T_i} = W_T^P h_{i, T}^{[\text{REP}]} + b_T^P \in \mathbb{R}^d \quad (11)$$

$$o_{T_i} = W_T^O h_{i, T}^{[\text{REP}]} + b_T^O \in \mathbb{R}^d \quad (12)$$

4.2.2 Image Artefact Originated Workflow Signals

We define $h_{i, F}^{[\text{REP}]} = \text{MaxPooling}(H_{F_i})$. $h_{i, F}^{[\text{REP}]}$ is linearly projected by three separate Projection Heads: Input (W_F^I, b_F^I), Process (W_F^P, b_F^P) and Output (W_F^O, b_F^O), to obtain the following Workflow Signals:

$$i_{F_i} = W_F^I h_{i,F}^{[\text{REP}]} + b_F^I \in \mathbb{R}^d \quad (13)$$

$$p_{F_i} = W_F^P h_{i,F}^{[\text{REP}]} + b_F^P \in \mathbb{R}^d \quad (14)$$

$$o_{F_i} = W_F^O h_{i,F}^{[\text{REP}]} + b_F^O \in \mathbb{R}^d \quad (15)$$

4.2.3 Document Artefact Originated Workflow Signals

The encoded $[\text{CLS}]_T$ representations of each text elements and the MaxPooled representations of each image elements of D_i , $\{h_{q,F \vee T}^{[\text{REP}]}\}_q \in H_{D_i}$ are retrieved and averaged into $h_{i,D}^{[\text{REP}]} \in \mathbb{R}^d$. The resulting vector is linearly projected by three separate Projection Heads: Input (W_D^I, b_D^I), Process (W_D^P, b_D^P) and Output (W_D^O, b_D^O), to obtain the following Workflow Signals:

$$i_{D_i} = W_D^I h_{i,D}^{[\text{REP}]} + b_D^I \in \mathbb{R}^d \quad (16)$$

$$p_{D_i} = W_D^P h_{i,D}^{[\text{REP}]} + b_D^P \in \mathbb{R}^d \quad (17)$$

$$o_{D_i} = W_D^O h_{i,D}^{[\text{REP}]} + b_D^O \in \mathbb{R}^d \quad (18)$$

5 Decoding Intention

We define a Workflow Intention γ as a triple of Input, Process and Output Workflow Signals:

$$\gamma = (i_\gamma, p_\gamma, o_\gamma) \quad (19)$$

We define the Workflow Intention Set of a set of Business Artefacts \mathcal{A} as a set of Workflow Intentions $\Gamma = \{\gamma_i\}_i$. The goal is to generate the Workflow Intention Set, i.e. Workflow Intention object(s), from a contextual representation of all the Business Artefacts. To do so we employ an encoder-decoder architecture described as follows.

5.1 Intra-Modality Attention

Across multiple Business Artefacts $\mathcal{A}_{m_k} = \{a_{m_k,i}\}_i$ of the same modality m_k , the encoded sequences $\{E_{m_k,i}^{\text{enc}}\}_i$ are concatenated, encoded by the self-attention based encoder **Encoder** $_{m_k}^{\text{intra}}$ into $H_{\mathcal{A}_{m_k}}^{\text{intra}}$. An encoded [REP] token representation $h_{\mathcal{A}_{m_k}}^{\text{intra},[\text{REP}]}$ is computed, linearly projected into the unified space by ($W_{m_k}^{\text{intra},u}, b_{m_k}^{\text{intra},u}$) then by three Projection Heads: Input ($W_{m_k}^{\text{intra},I}, b_{m_k}^{\text{intra},I}$), Process ($W_{m_k}^{\text{intra},P}, b_{m_k}^{\text{intra},P}$) and Output ($W_{m_k}^{\text{intra},O}, b_{m_k}^{\text{intra},O}$).

5.1.1 Artefact vectors Aggregation

For a given modality $m_k \in \{T, F, D\}$, let $\mathcal{A}_{m_k} = \{a_{m_k,i}\}_i$ be a set of Business Artefacts of this modality and $\{E_{m_k,i}^{\text{enc}}\}_i$ be the set of encoded tensors of these Business Artefacts.

$\forall i, E_{m_k,i}^{\text{enc}} \in \mathbb{R}^{d_{m_k} \times L_{m_k,i}}$ where $L_{m_k,i}$ is the number of encoded vectors of $a_{m_k,i}$.

$$E_{\mathcal{A}_{m_k}} = \text{Concat}(\{E_{m_k,i}^{\text{enc}}\}_i) \in \mathbb{R}^{d_{m_k} \times L_{\mathcal{A}_{m_k}}} \text{ where } L_{\mathcal{A}_{m_k}} = \sum_{i=1}^{|\mathcal{A}_{m_k}|} L_{m_k,i} \quad (20)$$

5.1.2 Intra-Modality Encoder and Signals

To capture contextual dependencies, we pass $E_{\mathcal{A}_{m_k}}$ through the encoder **Encoder** $_{m_k}^{\text{intra}}$ to calculate $E_{\mathcal{A}_{m_k}}^{\text{intra}}$ which is linearly projected by $(W_{m_k}^{\text{intra},u}, b_{m_k}^{\text{intra},u})$ to obtain $H_{\mathcal{A}_{m_k}}^{\text{intra}} \in \mathbb{R}^{d \times L_{\mathcal{A}_{m_k}}}$.

The representative encoded [REP] token representation of \mathcal{A}_{m_k} is computed as $h_{\mathcal{A}_{m_k}}^{\text{intra},[\text{REP}]} = \text{MaxPooling}(H_{\mathcal{A}_{m_k}}^{\text{intra}})$ and linearly projected by the three separate Projection Heads: Input $(W_{m_k}^{\text{intra},I}, b_{m_k}^{\text{intra},I})$, Process $(W_{m_k}^{\text{intra},P}, b_{m_k}^{\text{intra},P})$ and Output $(W_{m_k}^{\text{intra},O}, b_{m_k}^{\text{intra},O})$, to obtain the following Workflow Signals:

$$i_{\mathcal{A}_{m_k}} = W_{m_k}^{\text{intra},I} h_{\mathcal{A}_{m_k}}^{\text{intra},[\text{REP}]} + b_{m_k}^{\text{intra},I} \in \mathbb{R}^d \quad (21)$$

$$p_{\mathcal{A}_{m_k}} = W_{m_k}^{\text{intra},P} h_{\mathcal{A}_{m_k}}^{\text{intra},[\text{REP}]} + b_{m_k}^{\text{intra},P} \in \mathbb{R}^d \quad (22)$$

$$o_{\mathcal{A}_{m_k}} = W_{m_k}^{\text{intra},O} h_{\mathcal{A}_{m_k}}^{\text{intra},[\text{REP}]} + b_{m_k}^{\text{intra},O} \in \mathbb{R}^d \quad (23)$$

5.2 Inter-Modality Fusion Attention

Considering $\mathcal{A} = \{\mathcal{A}_{m_k}\}_k$ a set of Business Artefacts grouped by modality. From now on, we consider $\mathcal{A}_T, \mathcal{A}_F, \mathcal{A}_D$ sets of text, image and document Business Artefacts respectively.

5.2.1 Inter-Modality vectors Aggregation

We form a combined matrix H^{inter} by concatenating the intra-modality encoder outputs $\{H_{\mathcal{A}_{m_k}}^{\text{intra}}\}_k$ column-wise:

$$H^{\text{inter}} = \text{Concat}(\{H_{\mathcal{A}_{m_k}}^{\text{intra}}\}_k) \in \mathbb{R}^{d \times L_{\mathcal{A}}} \text{ where } L_{\mathcal{A}} = \sum_{k=1}^{|\mathcal{A}|} L_{\mathcal{A}_{m_k}} \quad (24)$$

5.2.2 Fusion Encoder

We pass H^{inter} through an encoder network **Encoder** $_{\text{fusion}}$ to calculate $H^{\text{fusion}} \in \mathbb{R}^{d \times L_{\mathcal{A}}}$. We currently do not employ projection heads to compute the Workflow Signals out of the fused representation of the Business Artefacts, as the fusion encoder is trained on Workflow Intention generation and not Workflow Signal extraction, as described later in the paper.

5.3 Intention Decoder

The decoder generates vectors based on the context computed from the artefacts. Each generated vector is projected into Workflow Signals i_γ, o_γ and p_γ , defining a Workflow Intention object γ as an element of the Workflow Intention Set Γ . It is made of N_{decoder} layers. Each layer is composed of a block of n_{decoder} masked self-attention heads coupled with a LayerNorm block, followed by a block of n_{decoder} cross-attention heads coupled with a LayerNorm block.

5.3.1 Generation loop

We initialize a decoded sequence S_0^{dec} with a [BOS] token embedding representation $\mathbf{E}_{\text{fusion}}(\text{[BOS]})$. At iteration t , in each decoder layer, S_t^{dec} is first encoded through the masked multi-head self-attention heads, then attends to the fusion encoder’s multimodal Business Artefact context H^{fusion} via the cross-attention heads. The output sequence encoded by all the layers is denoted as \tilde{S}_t^{dec} . The last vector of the sequence, denoted by $\tilde{s}_{t,-1}^{\text{dec}} \in \mathbb{R}^d$ is linearly projected by (W_γ, b_γ) to produce $\tilde{\gamma}_t \in \mathbb{R}^d$. We introduce two stopping mechanisms below: the Stopping Head and the Stopping Criteria. The Stopping Head acts as a first layer to stop the generation based the latest computed context. The Stopping Criteria stops the generation based on the latest generated Workflow Intention object. If the Stopping Head described below suggests to accept the generation, $\tilde{\gamma}_t \in \mathbb{R}^d$ is linearly projected by three separate Projection Heads: Input (W_γ^I, b_γ^I) , Process (W_γ^P, b_γ^P) and Output (W_γ^O, b_γ^O) to obtain the following Workflow Signals:

$$i_{\gamma_t} = W_\gamma^I \tilde{\gamma}_t + b_\gamma^I \in \mathbb{R}^d \quad (25)$$

$$p_{\gamma_t} = W_\gamma^P \tilde{\gamma}_t + b_\gamma^P \in \mathbb{R}^d \quad (26)$$

$$o_{\gamma_t} = W_\gamma^O \tilde{\gamma}_t + b_\gamma^O \in \mathbb{R}^d \quad (27)$$

These projections produce the Intention object:

$$\gamma_t = (i_{\gamma_t}, p_{\gamma_t}, o_{\gamma_t}) \quad (28)$$

If the Stopping Criteria described below suggest to accept and continue the generation, we start iteration $t + 1$ with:

$$S_{t+1}^{\text{dec}} = \text{Concat}(S_t^{\text{dec}}, \tilde{\gamma}_t) \quad (29)$$

Let t_f be the last iteration that passed the Stopping Mechanisms, we have:

$$\Gamma = \{\gamma_t\}_{t=1}^{t_f} \quad (30)$$

5.3.2 Stopping Mechanisms

Stopping Head We define the Stopping Head as

$$\text{MLP}_{\text{stop}} = \text{MLP}(\text{ReLU}, n_{\text{stop}}, (W_{\text{stop},i}, b_{\text{stop},i})_{i=1}^{n_{\text{stop}}}, (0, 1))$$

where 0 denotes the “Stop” class to stop the generation and 1 the “Accept” class to accept the current generation a priori. The intuition is to decide if the current generated sequence of Workflow Intentions, attended with the Business Artefacts context, is complete or not.

$$\forall t > 1, \text{MLP}_{\text{stop}}(\tilde{s}_{t,-1}^{\text{dec}}) = \delta_t^{\text{head}} \in \{0, 1\} \quad (31)$$

$$\text{With } \delta_t^{\text{head}} = \begin{cases} 1 & \text{if } \mathbb{P}_t(\text{“Accept”}) > 0.5 \\ 0 & \text{else} \end{cases} \quad \text{and } \delta_1^{\text{head}} = 1 \quad (32)$$

Stopping Criteria We define the Redundant Stopping Criterion as

$$\forall t > 1, \delta_t^{\text{sim}} = \begin{cases} 1 & \text{if } \frac{1}{3} \left(\frac{\langle i_{\gamma_{t'}}, i_{\gamma_t} \rangle}{\|i_{\gamma_{t'}}\| \|i_{\gamma_t}\|} + \frac{\langle p_{\gamma_{t'}}, p_{\gamma_t} \rangle}{\|p_{\gamma_{t'}}\| \|p_{\gamma_t}\|} + \frac{\langle o_{\gamma_{t'}}, o_{\gamma_t} \rangle}{\|o_{\gamma_{t'}}\| \|o_{\gamma_t}\|} \right) < \tau^{\text{sim}} \quad \forall t' \in \llbracket 1, t-1 \rrbracket \\ 0 & \text{else} \end{cases} \quad (33)$$

$$\text{With } \delta_1^{\text{sim}} = 1 \text{ and } \tau^{\text{sim}} \in [0, 1] \quad (34)$$

At step t , $\delta_t^{\text{sim}} = 1$ indicates to continue the generation whereas $\delta_t^{\text{sim}} = 0$ indicates to stop the generation. The intuition is to refuse γ_t and stop the generation if the generated Workflow Intention at step t is too similar to one of the previously generated Workflow Intention.

We define the Hard Stopping Criterion by t_{max} such that if $t > t_{\text{max}}$ the generation is stopped. This means that we constrain a user query to not include more than t_{max} distinct Workflow Intentions.

6 Training

6.1 Phase 1: Business Artefacts Encoding and Signal Extraction

We employ a two stage training regimen. First, we train each modality independently for each Business Artefact in each modality m_k where we have an **Encoder** $_{m_k}$ that is finetuned and $(W_{m_k}^u, b_{m_k}^u)$, $(W_{m_k}^I, b_{m_k}^I)$, $(W_{m_k}^P, b_{m_k}^P)$, $(W_{m_k}^O, b_{m_k}^O)$ are trained by passing $N_{\text{artefact}, m_k}^{(1.1)}$ Business Artefacts for each modality.

For all modalities m_k , $\mathcal{A}_{m_k}^{(1.1)} = \{a_{m_k, i}^{(1.1)}\}_{i=1}^{N_{\text{artefact}, m_k}^{(1.1)}}$ denotes the set of training Business Artefacts for modality m_k at Stage 1 of Phase 1 ($N_{\text{artefact}, m_k}^{(1.1)} = |\mathcal{A}_{m_k}^{(1.1)}|$).

Then, we continue training each modality independently over the intra-modality layers so that for all m_k , **Encoder** $_{m_k}^{\text{intra}}$ is finetuned and $(W_{m_k}^{\text{intra}, u}, b_{m_k}^{\text{intra}, u})$, $(W_{m_k}^{\text{intra}, I}, b_{m_k}^{\text{intra}, I})$, $(W_{m_k}^{\text{intra}, P}, b_{m_k}^{\text{intra}, P})$, $(W_{m_k}^{\text{intra}, O}, b_{m_k}^{\text{intra}, O})$ are trained by passing $N_{\text{set}, m_k}^{(1.2)}$ Business Artefact sets for each modality.

For all modalities m_k , we provide $\{\mathcal{A}_{m_k, j}^{(1.2)}\}_{j=1}^{N_{\text{set}, m_k}^{(1.2)}} = \{\{a_{m_k, i, j}^{(1.2)}\}_{i=1}^{|\mathcal{A}_{m_k, j}^{(1.2)}|}\}_{j=1}^{N_{\text{set}, m_k}^{(1.2)}}$ denoting the sets of training Business Artefacts for modality m_k at Stage 2 of Phase 1.

In total, $\sum_{m_k} N_{\text{artefact}, m_k}^{(1.1)}$ Business Artefacts are provided in stage 1 and $\sum_{m_k} \sum_{j=1}^{N_{\text{set}, m_k}^{(1.2)}} |\mathcal{A}_{m_k, j}^{(1.2)}|$ in stage 2.

6.1.1 Classification Tasks for i, o and p

We build ground truth data based on three sets of text elements I_g, P_g and O_g :

- I_g : elements that serve as input Workflow Signals within the Business Artefacts.
- P_g : elements that relate to transformations or Processes within the Business Artefacts.
- O_g : elements that describe expected output Workflow Signals within the Business Artefacts.

Each projected vector i, p and o is associated with a ground truth representation over its set denoted by:

$$C_x^* \in \mathbb{R}^{|X_g| \times (M+2)}, \quad \text{where } (x, X) \in \{(i, I), (p, P), (o, O)\}, M \in \mathbb{N}^* \quad (35)$$

Each projected vector $x \in \{i, o, p\}$ is passed through a dedicated MLP_X to predict discrete counts for each class in X_g up to a maximum count M .

Each classifier ($\text{MLP}_I, \text{MLP}_P, \text{MLP}_O$) outputs a set of logits for each class:

$$\hat{C}_x \in \mathbb{R}^{|X_g| \times (M+2)}, \quad \text{where } (x, X) \in \{(i, I), (p, P), (o, O)\} \quad (36)$$

Each row i of \hat{C}_x represents the unnormalized logits for predicting the count class c of the corresponding element $x_{g,i}$ of X_g where:

- $c=0$: $x_{g,i}$ not present
- $c \in \llbracket 1, M \rrbracket$: $x_{g,i}$ referenced in plural form with known exact count c .
- $c = M + 1$: $x_{g,i}$ referenced in plural form, but exact count is unknown.

6.1.2 Loss

For each class indexed by $k \in \llbracket 1, |X_g| \rrbracket$, given a ground-truth count class $c_k^* \in \llbracket 0, M + 1 \rrbracket$, we apply a categorical cross-entropy loss:

$$\mathcal{L}_{X,k} = - \sum_{m=0}^{M+1} \delta_m(c_k^*) \log(\text{softmax}(\hat{C}_x[k])[m]) \quad \text{where } \delta_m : x \mapsto \begin{cases} 1 & \text{if } x = m \\ 0 & \text{else} \end{cases} \quad (37)$$

The total loss for each head is computed as:

$$\mathcal{L}_X = \frac{1}{|X_g|} \sum_{k=1}^{|X_g|} \mathcal{L}_{X,k} \quad (38)$$

The overall loss is the sum over the three heads:

$$\mathcal{L}_{\text{signal}} = \text{bound}(\mathcal{L}_I + \mathcal{L}_P + \mathcal{L}_O, \lambda, \mu) \quad (39)$$

With the bounding function $\text{bound}(\mathcal{L}, \lambda, \mu) = \frac{1}{1+e^{-\lambda(\mathcal{L}-\mu)}}$, $\mu \in [0, 1]$ and $\lambda > 0$.

Stage 1 $\forall m_k, \mathcal{A}_{m_k}^{(1.1)} = \{a_{m_k,i}^{(1.1)}\}_{i=1}^{N_{\text{artefact},m_k}^{(1.1)}}$ denotes the set of training Business Artefacts for modality m_k at Stage 1 of Phase 1. Each Business Artefact is tokenized, encoded, projected then classified.

Stage 2 At this stage, all the elements of stage 1 are frozen.

$\forall m_k, \{\mathcal{A}_{m_k,j}^{(1.2)}\}_{j=1}^{N_{\text{set},m_k}^{(1.2)}} = \{\{a_{m_k,i,j}^{(1.2)}\}_{i=1}^{|\mathcal{A}_{m_k,j}^{(1.2)}|}\}_{j=1}^{N_{\text{set},m_k}^{(1.2)}}$ denotes the sets of training Business Artefacts for modality m_k at Stage 2 of Phase 1. Each Business Artefact of each set is tokenized and encoded, then across each set the encoded Business Artefacts are concatenated, encoded, projected then classified. We define the ground truth over an Business Artefact set as $C_{x,\mathcal{A}_{m_k,j}^{(1.2)}}^*, \forall m_k, \forall j \in \llbracket 1, N_{\text{set},m_k}^{(1.2)} \rrbracket$.

6.2 Phase 2: Decoding Intention

In this phase, all the elements from Phase 1 are frozen. The training data for this phase is $\mathcal{A}^{(2.2)} = \{\mathcal{A}_q^{(2.2)}\}_{q=1}^{N^{(2.2)}}$ where $N^{(2.2)}$ denotes the number of samples. Each sample is such that $\mathcal{A}_q^{(2.2)} = \{\mathcal{A}_{q,m_k}^{(2.2)}\}_{m_k}$ where $\mathcal{A}_{q,m_k}^{(2.2)}$ is a set of Business Artefacts of modality m_k . For each sample, across each modality, across each set, each Business Artefact is tokenized and encoded. Across each set, encoded Business Artefacts are concatenated and encoded by the intra-modality encoder and projected in the d dimension. Across each modality, the intra-modality encoded vectors are concatenated and the resulting sequence of vectors by **Encoder**_{fusion}. The decoding loop starts, provided with the entire context of the sample $\mathcal{A}_q^{(2.2)}$ from **Encoder**_{fusion}. **Decoder**_{Intention} is producing a sequence $\hat{\Gamma} = \{\hat{\gamma}_t\}_{t=1}^{|\hat{\Gamma}|}$. We train the system by classifying $\hat{i}_{\gamma_t}, \hat{p}_{\gamma_t}, \hat{o}_{\gamma_t}$ of each γ_t using $\text{MLP}_I^\gamma, \text{MLP}_O^\gamma, \text{MLP}_P^\gamma$ and compute the loss (described below) over a ground truth $i_{\gamma_t}^*, p_{\gamma_t}^*$ and $o_{\gamma_t}^*$ expressed over the sets I_g, P_g, O_g as done for Stage 1.

Stopping Head and Intention Generation Losses Given a generated Workflow Intention: $\hat{\Gamma} = \{\hat{\gamma}_t\}_{t=1}^{\hat{t}_f}$ and a ground truth $\Gamma^* = \{\gamma_t^*\}_{t=1}^{t_f^*}$:

$$\forall \gamma \in \hat{\Gamma} \cup \Gamma^*, \gamma = (i_\gamma, p_\gamma, o_\gamma) \quad (40)$$

We use $\mathcal{L}_{\text{signal}}$ defined previously and introduce a threshold τ_γ to consider two Workflow Intentions γ_1, γ_2 matching if and only if $\mathcal{L}_{\text{signal}}(\gamma_1, \gamma_2) \leq \tau_\gamma$.

We introduce the Coverage measure between Γ^* and $\hat{\Gamma}$ as:

$$\text{Coverage}_\Gamma(\Gamma^*, \hat{\Gamma}) = \frac{1}{t_f^*} \sum_{t=1}^{\hat{t}_f} c_{\gamma,t} \quad (41)$$

$$\text{where } c_{\gamma,t} = \begin{cases} 1 & \text{if } \min_{\gamma \in \hat{\Gamma}} (\{\mathcal{L}_{\text{signal}}(\gamma_t^*, \gamma)\}) < \tau_\gamma \\ 0 & \text{else} \end{cases} \quad (42)$$

We define

- For overlength: $\Delta_\Gamma^+ = \max(0, \hat{t}_f - t_f^*)$
- For underlength: $\Delta_\Gamma^- = \max(0, t_f^* - \hat{t}_f)$

We define the Workflow Intention sequence loss in terms of coverage, overlength and underlength, such as

$$\mathcal{L}_{\text{sequence}} = 1 - \left[\alpha_{\Gamma,c} \cdot \text{Coverage}_\Gamma + \alpha_{\Gamma,o} \frac{1}{1 + \Delta_\Gamma^+} + \alpha_{\Gamma,u} \frac{1}{1 + \Delta_\Gamma^-} \right] \quad (43)$$

where $\alpha_{\Gamma,c} + \alpha_{\Gamma,o} + \alpha_{\Gamma,u} = 1$,
 $0 \leq \alpha_{\Gamma,c} \leq 1$, $0 \leq \alpha_{\Gamma,o} \leq 1$ and $0 \leq \alpha_{\Gamma,u} \leq 1$

We define the Workflow Intention contrastive loss to encourage diverse Workflow Intention generation as:

$$\mathcal{L}_{\text{contrastive}} = \begin{cases} \frac{2}{\hat{t}_f(\hat{t}_f-1)} \sum_{m=1}^{\hat{t}_f} \sum_{n=m+1}^{\hat{t}_f} e^{-(\|i_{\gamma_m} - i_{\gamma_n}\|^2 + \|p_{\gamma_m} - p_{\gamma_n}\|^2 + \|o_{\gamma_m} - o_{\gamma_n}\|^2)} & \text{if } \hat{t}_f > 1 \\ 0 & \text{otherwise} \end{cases} \quad (44)$$

with $\forall i, \gamma_i \in \hat{\Gamma}$.

We define the Stopping Head loss as:

$$\mathcal{L}_{\text{head}} = -\frac{1}{\max(\hat{t}_f, t_f^*)} \sum_{t=1}^{\max(\hat{t}_f, t_f^*)} [\delta_t^{\text{head}} \log(\mathbb{P}_t(\text{"Accept"})) + (1 - \delta_t^{\text{head}}) \log(1 - \mathbb{P}_t(\text{"Accept"}))] \quad (45)$$

$$\text{with } \begin{cases} \mathbb{P}_t(\text{"Accept"}) = 0 & \forall t > t_f^* \text{ if } t_f^* > \hat{t}_f \\ \delta_t^{\text{head}} = 0 & \forall t > t_f^* \text{ if } t_f^* < \hat{t}_f \end{cases}$$

We define the Workflow Intention Loss as:

$$\mathcal{L}_{\text{Intention}} = \mathcal{L}_{\text{head}} + \mathcal{L}_{\text{contrastive}} + \mathcal{L}_{\text{sequence}} \quad (46)$$

7 Computational Complexity

Parameter count analysis reveals significant overhead:

- Text encoder: $24 \text{ layers} \times [(16 \text{ heads} \times 3 \text{ matrices} \times 64 \times 1024) + (1024 \times 1024) + (1024 \times 4096 \times 2)] \approx 300\text{M}$ parameters
- Image encoder: $588 \times 3200 + 45 \text{ layers} \times [(25 \text{ heads} \times 3 \text{ matrices} \times 128 \times 3200) + (3200 \times 3200) + (3200 \times 12800 \times 2)] \approx 5.5\text{B}$ parameters
- Document encoder:
 - Text Encoder $\approx 300\text{M}$
 - Image Encoder $\approx 5.5\text{B}$
 - Document Encoder same as Text Encoder $\approx 300\text{M}$ $\approx 6\text{B}$ parameters
- Fusion Encoder: $24 \text{ layers} \times [(128 \text{ heads} \times 3 \text{ matrices} \times 8 \times 1024) + (1024 \times 1024) + (1024 \times 65536 \times 2)] \approx 3.3\text{B}$ parameters
- Intention Decoder: $24 \text{ layers} \times [2 \times ((128 \text{ heads} \times 3 \text{ matrices} \times 8 \times 1024) + (1024 \times 1024)) + (1024 \times 65536 \times 2)] \approx 3.5\text{B}$ parameters

- Projection Heads:
 - Text Unifier: 1024×1024
 - Text Input, Output, Process Projections: $3 \text{ heads} \times 1024 \times 1024$
 - Image Unifier: 3200×1024
 - Image Input, Output, Process Projections: $3 \text{ heads} \times 1024 \times 1024$
 - Document Unifier: $3200 \times 1024 + 1024 \times 1024$
 - Document Input, Output, Process Projections: $3 \text{ heads} \times 1024 \times 1024$
 - Decoder: 1024×1024
 - Decoder Input, Output, Process: $3 \text{ heads} \times 1024 \times 1024$
- 2 [per Artefact and Intra modality] $\times [1024 \times 1024 + 3 \times 1024 \times 1024 + 3200 \times 1024 + 3 \times 1024 \times 1024 + 1024 \times 1024 + 3 \times 1024 \times 1024] + [3200 \times 1024] + [1024 \times 1024 + 3 \times 1024 \times 1024] \approx 38\text{M}$ parameters
- MLPs: Assuming $|X_g| \approx 10^5, X \in \{\text{I, P, O}\}$,
 - 2 Signal Classifications [per Artefact and Intra modality] $\times 3$ modalities
 - 1 Attention Signal Classification
 - 1 Stopping Mechanism
- $7 \times 3 \text{ heads} \times [4096 \times 1024 + 4096 \times 10^5] + [4096 \times 1024 + 4096 \times 2] \approx 8.7\text{B}$ parameters

This results in an approximate total of 27.5 billion parameters, excluding the tokenizer and token encoder parameters, which are provided out of the box.

There are challenges due to the computational complexity of the system. The Workflow Intention framework exhibits $\mathcal{O}(n^2d + nd^2)$ complexity in the attention mechanisms, where n represents the sequence length and d represents the embedding dimension. This quadratic scaling becomes problematic in the inter-modality fusion encoder, where $n = 1 + \sum_{i=1}^{|\mathcal{A}|} L_{\mathcal{A}_{m_k}}$.

The document encoder’s representational overhead is particularly significant, requiring $2(L_{D_{p,i}}^T + L_{D_{p,i}}^F)$ vectors of dimension d_T for each document page. To optimize computational efficiency, the sequence length n can be reduced using sparsification techniques like Longformer (sliding window attention, Beltagy et al. [9]) or Linformer (low-rank approximation, Wang et al. [10]). The hidden dimension d in intermediate layers can be decreased to lower the $\mathcal{O}(d^2)$ cost in the encoders. Additionally, compressing document representations via pre-processing pipelines would reduce the number of stored vectors per page and improve memory efficiency while preserving essential information. The computational complexities incurred by the length of the decoded sequence are negligible as the number of Intentions is typically bounded between 1 and 5.

8 Conclusion

In this paper, we have introduced Workflow Intention, a comprehensive framework for identifying and encoding Process objectives within complex business environments. Our approach addresses the fundamental challenge of interpreting and leveraging Process documentation through a systematic methodology that interprets and aligns Input, Process and Output Workflow Signals from diverse Business Artefacts. The mathematical framework we developed formalizes these Workflow Signals as vectors and Workflow Intentions as tensors, providing a rigorous foundation for understanding Process objectives.

The multimodal generative system we developed demonstrates the practical applicability of our framework, successfully processing various types of Business Artefacts through Modality-Specific Encoding, Intra-Modality Attention, Inter-Modality Fusion Attention and Intention Decoding. Our hierarchical encoder methodology effectively generates Workflow Intention from Workflow Signal across modalities. This work enables organizations to rapidly implement supervised automation and evolve legacy processes into efficient AI-enhanced Workflows enriched with best practice.

References

- [1] Fagnoni, T., Mesbah, B., Altin, M., and Kingston, P. (2024). Opus: A Large Work Model for Complex Workflow Generation.
- [2] Vaswani, A. (2017). Attention is All You Need. *Advances in Neural Information Processing Systems*, 30, 5998–6008.
- [3] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach.
- [4] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. (2023). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.
- [5] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.
- [6] Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., and Kiela, D. (2022). FLAVA: A Foundational Language And Vision Alignment Model.
- [7] Dai, W., Lee, N., Wang, B., Yang, Z., Liu, Z., Barker, J., Rintamaki, T., Shoeybi, M., Catanzaro, B., and Ping, W. (2024). NVLM: Open Frontier-Class Multimodal LLMs.
- [8] Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Cui, E., Zhu, J., Ye, S., Tian, H., Liu, Z., Gu, L., Wang, X., Li, Q., Ren, Y., Chen, Z., Luo, J., Wang, J., et al. (2021). Expanding Performance Boundaries of Open-Source Multimodal Models with Model, Data, and Test-Time Scaling.
- [9] Beltagy, I., Peters, M., and Cohan, A. (2020). Longformer: The Long-Document Transformer.
- [10] Wang, S., Li, B., Khabsa, M., Fang, H., and Ma, H. (2020). Linformer: Self-Attention with Linear Complexity.

A Appendix

We describe the models and mechanisms we employ, including the backbone architectures and parametrization, as well as the mathematical interpretation of our framework.

A.1 Classic Computational Mechanisms

A.1.1 softmax

The softmax function is defined such that:

$$\forall \mathbf{X} = (x_i)_{i=1}^{d_X} \in \mathbb{R}^{d_X}, \text{softmax}(\mathbf{X}) = \left(\frac{e^{x_i}}{\sum_{j=1}^{d_X} e^{x_j}} \right)_{i=1}^{d_X} \in \mathbb{R}^{d_X} \quad (47)$$

By default, for a matrix $\mathbf{M} \in \mathbb{R}^{d \times n}$, $\text{softmax}(\mathbf{X})$ denotes the row-wise application of the softmax function i.e.

$$\text{softmax}(\mathbf{M}) = (\text{softmax}(\mathbf{M}[i,:]))_{i=1}^d \quad (48)$$

A.1.2 LayerNorm

The LayerNorm mechanism is described by a parameter ϵ and two learnable parameters (γ, β) such that:

$$\begin{aligned} \forall \mathbf{X} &= (x_i)_{i=1}^{d_X} \in \mathbb{R}^{d_X}, \\ \text{LayerNorm}(\mathbf{X}) &= \frac{\mathbf{X} - \mu}{\sigma} \cdot \gamma + \beta \end{aligned} \quad (49)$$

where $\mu = \frac{1}{d_X} \sum_{i=1}^{d_X} x_i$, $\sigma_i = \sqrt{\frac{1}{d_X} \sum_{i=1}^{d_X} (x_i - \mu)^2 + \epsilon}$

A.1.3 Linear Projection

A linear projection mechanism \mathbf{L} is described by a learnable weight matrix and bias term $(\mathbf{W}_L, \mathbf{b}_L)$ with $\mathbf{W}_L \in \mathbb{R}^{d_L \times d_X}$, $\mathbf{b}_L \in \mathbb{R}^{d_L}$ such that

$$\forall y \in \mathbb{N}^*, \forall \mathbf{X} \in \mathbb{R}^{d_X \times y}, \mathbf{L}(\mathbf{X}) = \mathbf{W}_L \mathbf{X} + \mathbf{b}_L \in \mathbb{R}^{d_L \times y} \quad (50)$$

The bias is broadcast across all columns of $\mathbf{W}_L \mathbf{X}$.

A.1.4 MLP

A Multi-Layer-Perceptron MLP is described by n_{MLP} layers of Linear Projection and activation function: $((f_{\text{act},i}, \mathbf{W}_i, \mathbf{b}_i))_{i=1}^{n_{\text{MLP}}}$, such that

$$\begin{aligned} \forall y \in \mathbb{N}^*, \forall \mathbf{X} \in \mathbb{R}^{d_X \times y}, \\ \text{MLP}(\mathbf{X}) &= f_{\text{act},n_{\text{MLP}}}(\mathbf{W}_{n_{\text{MLP}}-1} f_{\text{act},n_{\text{MLP}}-1}(\dots f_{\text{act},1}(\mathbf{W}_1 \mathbf{X} + \mathbf{b}_1)\dots) + \mathbf{b}_{n_{\text{MLP}}}) \end{aligned} \quad (51)$$

$\text{MLP}_I, \text{MLP}_O, \text{MLP}_P$; $\text{MLP}_I^{\text{intra}}, \text{MLP}_O^{\text{intra}}, \text{MLP}_P^{\text{intra}}$; $\text{MLP}_I^\gamma, \text{MLP}_O^\gamma, \text{MLP}_P^\gamma$; MLP_{stop}
 These MLP networks are such that $((\text{softmax}, \text{W}_1^{\text{MLP}_X}, \text{b}_1^{\text{MLP}_X}), (\text{Id}, \text{W}_2^{\text{MLP}_X}, \text{b}_2^{\text{MLP}_X}))$ with an inner dimension of 4096 i.e. $\text{W}_1^{\text{MLP}_X} \in \mathbb{R}^{4096 \times 1024}$ and $\text{W}_2^{\text{MLP}_X} \in \mathbb{R}^{|X_g| \times 4096}$, $X \in \{I, P, O\}$.

A.1.5 FFN

A Feed-Forward-Network FFN is described by n_{FFN} layers of Linear Projection and activation function $((f_{\text{act},i}, \text{W}_i, \text{b}_i))_{i=1}^{n_{\text{FFN}}}$, such that

$$\forall y \in \mathbb{N}^*, \forall (X_i)_{i=1}^y \in \mathbb{R}^{d_X \times y},$$

$$\text{FFN}(X) = \left(f_{\text{act},n_{\text{FFN}}}(\text{W}_{n_{\text{FFN}}} f_{\text{act},n_{\text{FFN}}-1}(\dots f_{\text{act},1}(\text{W}_1 X_i + \text{b}_1)\dots) + \text{b}_{n_{\text{FFN}}}) \right)_{i=1}^y \quad (52)$$

A.2 Tokenizers and Token Encoders

Text We employ the RoBERTa-large (Liu et al. [3]) Byte-Level Byte Pair Encoding (BPE) tokenizer which has a 50 265 token vocabulary including the start-of-sequence token $\langle \mathbf{s} \rangle$ which we label as $[\text{CLS}]_T$ token. The token encoder embeds the tokens into $d_T = \mathbb{R}^{1024}$ as well as positional embeddings and sums both representations.

Image We employ the tiling, unshuffling and flattening method of InternViT-6B-448px-V1.5 presented by Dosovitskiy et al. [5] and expended in InternVL by Chen et al. [8]. Each image $F \in \mathbb{R}^{c \times h \times w}$ (with $c = 3$) is resized in an optimal ratio $r^*(\tilde{t}) := w^*/h^*$ such that when divided into tiles $\{\tilde{t}_k\}_{k=1}^{N_{\tilde{t}}}$ where $\forall k, \tilde{t}_k \in \mathbb{R}^{c \times \tilde{t} \times \tilde{t}}$ with $N_{\tilde{t}} = h^* \times w^*/\tilde{t}^2$, $N_{\tilde{t}} < n_{\text{max}}$. For InternViT-6B-448px-V1.5, $\tilde{t} = 448$, $n_{\text{max}} = 12$. If $N_{\tilde{t}} > 1$, a thumbnail tile, which is a resized version of the image to the target dimension \tilde{t} , is added to the sequence of tiles. Each tile \tilde{t}_k is then unshuffled 5 times by reducing the resolution dimensionality and increasing the number of channels, resulting in a sequence of patches $\{\tilde{p}_{k,i}\}_{i=1}^{N_{\tilde{p}}}$ where $\forall i, \tilde{p}_{k,i} \in \mathbb{R}^{c \times \tilde{p} \times \tilde{p}}$ with $\tilde{p} = 14$ and $N_{\tilde{p}} = 1024$ as $448 \times 448 \times 3 = (2^5 \times 2^5) \times (14 \times 14) \times 3 = 1024 \times (14 \times 14) \times 3 = 1024 \times 588$. Each of the 1024 patches is flattened to \mathbb{R}^{588} and linearly projected to \mathbb{R}^{3200} , thus $d_F = 3200$. The resulting sequence is added to learned positional embeddings from ViT (Dosovitskiy et al. [5]).

Document The Document tokenizer and token encoder process is fully described in the article, combining Text and Image tokenizers and token encoders.

A.3 Attention Mechanisms

A.3.1 Self-Attention

From [2], given an input sequence represented as a matrix $X \in \mathbb{R}^{d \times n}$, where d is the embedding dimension and n is the sequence length (number of tokens). Let $W^Q \in \mathbb{R}^{d_k \times d}$, $W^K \in \mathbb{R}^{d_k \times d}$, $W^V \in \mathbb{R}^{d_v \times d}$ be learnable weight matrices. The Queries, Keys and Values are computed as $Q = W^Q X \in \mathbb{R}^{d_k \times n}$, $K = W^K X \in \mathbb{R}^{d_k \times n}$, $V = W^V X \in \mathbb{R}^{d_v \times n}$.

The Attention is computed as

$$Z = V \cdot \text{softmax}(A) \in \mathbb{R}^{d_v \times n}, A = \frac{K^\top Q}{\sqrt{d_k}} \in \mathbb{R}^{n \times n} \quad (53)$$

We denote $Z = \text{SelfAttention}(X, W^Q, W^K, W^V)$

A.3.2 Masked Self-Attention

Following previous notation,

$$Z = V.\text{softmax}(A + M) \in \mathbb{R}^{d_v \times n}, M \in \mathbb{R}^{n \times n}, M[i, j] = \begin{cases} 0 & \text{if } i \geq j \\ -\infty & \text{otherwise} \end{cases} \quad (54)$$

A.3.3 Multi-Head Attention

Let H be the number of attention heads, W^O a learnable weight matrix. The Multi-Head Attention is computed as

$$\begin{aligned} Z &= W^O.\text{Concat}(\{\text{SelfAttention}(X, W_i^Q, W_i^K, W_i^V)\}_{i=1}^H) \in \mathbb{R}^{d_v \times n}, \\ \forall i, W_i^Q &\in \mathbb{R}^{\frac{d_k}{H} \times d}, W_i^K \in \mathbb{R}^{\frac{d_k}{H} \times d}, W_i^V \in \mathbb{R}^{\frac{d_v}{H} \times d}, W^O \in \mathbb{R}^{d_v \times d_v} \end{aligned} \quad (55)$$

A.3.4 Cross Attention

Given an input sequence represented as a matrix $X \in \mathbb{R}^{d \times n}$ and a context sequence represented as a matrix $Y \in \mathbb{R}^{d \times m}$. Let $W^Q \in \mathbb{R}^{d_k \times d}, W^K \in \mathbb{R}^{d_k \times d}, W^V \in \mathbb{R}^{d_v \times d}$ be learnable weight matrices. The Queries, Keys and Values are computed as $Q = W^Q X \in \mathbb{R}^{d_k \times n}, K = W^K Y \in \mathbb{R}^{d_k \times m}, V = W^V Y \in \mathbb{R}^{d_v \times m}$.

The Attention is computed as

$$Z = V.\text{softmax}(A) \in \mathbb{R}^{d_v \times n}, A = \frac{K^\top Q}{\sqrt{d_k}} \in \mathbb{R}^{m \times n} \quad (56)$$

We denote $Z = \text{CrossAttention}(X, Y, W^Q, W^K, W^V)$

A.4 Transformer Models

A.4.1 Encoder

Each **Encoder** we employ uses Multi-Head Self-Attention and is described by n_{Enc} layers and a dimension d . Each layer l is composed of:

1. Multi-Head Self-Attention:

- H_{Enc} Self-Attention heads, parametrized by $(W_i^{Q,l}, W_i^{K,l}, W_i^{V,l})_{i=1}^{H_{\text{Enc}}}$, each matrix being in $\mathbb{R}^{d_{H_{\text{Enc}}} \times d}$ where $d_{H_{\text{Enc}}} = d/H_{\text{Enc}}$
- A projection matrix $W^{O,l} \in \mathbb{R}^{d \times d}$

- #### 2. LayerNorm and Add:
- LayerNorm and the residual ‘‘Add’’ operation are applied element-wise across each token’s embedding dimension, i.e. column-wise when the input is structured as $d \times n$ where d is the embedding dimension and n is the sequence length.

3. **Feedforward Network (FFN):** $((\text{GELU}, W_1^{\text{FFN},l}, b_1^{\text{FFN},l}), (\text{Id}, W_2^{\text{FFN},l}, b_2^{\text{FFN},l}))$.
4. **LayerNorm and Add:** LayerNorm and the residual “Add” operation are applied element-wise across each token’s embedding dimension, i.e. column-wise when the input is structured as $d \times n$ where d is the embedding dimension and n is the sequence length.

Text Encoder_T We employ the RoBERTa-large (Liu et al. [3]) which has $n_{\text{Enc}} = 24$ and a final dimension $d_T = 1024$. In each layer l , the Multi-Head Self-Attention is made of $H_{\text{Enc}} = 16$ heads. The Feedforward Network is defined as $((\text{GELU}, W_1^{\text{FFN},l}, b_1^{\text{FFN},l}), (\text{Id}, W_2^{\text{FFN},l}, b_2^{\text{FFN},l}))$ with an inner dimension of 4096 i.e. $W_1^{\text{FFN},l} \in \mathbb{R}^{4096 \times 1024}$ and $W_2^{\text{FFN},l} \in \mathbb{R}^{1024 \times 4096}$.

Image Encoder_F We employ the InternViT-6B-448px-V1-5 (Dosovitskiy et al., [5]) which has $n_{\text{Enc}} = 45$ and a final dimension $d_F = 3200$. In each layer l , the Multi-Head Self-Attention is made of $H_{\text{Enc}} = 25$ heads. The Feedforward Network is defined as $((\text{GELU}, W_1^{\text{FFN},l}, b_1^{\text{FFN},l}), (\text{Id}, W_2^{\text{FFN},l}, b_2^{\text{FFN},l}))$ with an inner dimension of 12800 i.e. $W_1^{\text{FFN},l} \in \mathbb{R}^{12800 \times 3200}$ and $W_2^{\text{FFN},l} \in \mathbb{R}^{3200 \times 12800}$.

Document Encoder_D As for the text encoder, we employ the RoBERTa-large (Liu et al. [3]) which has $n_{\text{Enc}} = 24$ and a final dimension $d_T = 1024$. In each layer l , the Multi-Head Self-Attention is made of $H_{\text{Enc}} = 16$ heads. The Feedforward Network is defined as $((\text{GELU}, W_1^{\text{FFN},l}, b_1^{\text{FFN},l}), (\text{Id}, W_2^{\text{FFN},l}, b_2^{\text{FFN},l}))$ with an inner dimension of 4096 i.e. $W_1^{\text{FFN},l} \in \mathbb{R}^{4096 \times 1024}$ and $W_2^{\text{FFN},l} \in \mathbb{R}^{1024 \times 4096}$.

Encoder_T^{intra}, Encoder_F^{intra}, Encoder_D^{intra} These encoders are **Encoder_T**, **Encoder_F**, **Encoder_D** (trained in Phase 1 Stage 1), further trained in Phase 1 Stage 2.

A.4.2 Decoder

The **Decoder** we employ with an **Encoder** uses Masked Multi-Head Self-Attention and Multi-Head Cross Attention described by n_{Dec} layers and a dimension d . Each layer l is composed of:

1. **Masked Multi-Head Self-Attention:**
 - $H_{\text{Dec}}^{\text{self}}$ Self-Attention heads, parametrized by $(W_i^{Q,\text{self},l}, W_i^{K,\text{self},l}, W_i^{V,\text{self},l})_{i=1}^{H_{\text{Dec}}^{\text{self}}}$, each matrix being in $\mathbb{R}^{d_{H_{\text{Dec}}^{\text{self}}} \times d}$ where $d_{H_{\text{Dec}}^{\text{self}}} = d/H_{\text{Dec}}^{\text{self}}$
 - A projection matrix $W^{O,\text{self},l} \in \mathbb{R}^{d \times d}$
2. **LayerNorm and Add:** LayerNorm and the residual “Add” operation are applied element-wise across each token’s embedding dimension, i.e. column-wise when the input is structured as $d \times n$ where d is the embedding dimension and n is the sequence length.
3. **Multi-Head Cross Attention:**
 - $H_{\text{Dec}}^{\text{cross}}$ Cross Attention heads, parametrized by $(W_i^{Q,\text{cross},l}, W_i^{K,\text{cross},l}, W_i^{V,\text{cross},l})_{i=1}^{H_{\text{Dec}}^{\text{cross}}}$, each matrix being in $\mathbb{R}^{d_{H_{\text{Dec}}^{\text{cross}}} \times d}$ where $d_{H_{\text{Dec}}^{\text{cross}}} = d/H_{\text{Dec}}^{\text{cross}}$
 - A projection matrix $W^{O,\text{cross},l} \in \mathbb{R}^{d \times d}$

4. **LayerNorm and Add:** LayerNorm and the residual “Add” operation are applied element-wise across each token’s embedding dimension, i.e. column-wise when the input is structured as $d \times n$ where d is the embedding dimension and n is the sequence length.
5. **Feedforward Network (FFN):** $((\text{ReLU}, W_1^{\text{FFN},l}, b_1^{\text{FFN},l}), (\text{Id}, W_2^{\text{FFN},l}, b_2^{\text{FFN},l}))$.
6. **LayerNorm and Add:** LayerNorm and the residual “Add” operation are applied element-wise across each token’s embedding dimension, i.e. column-wise when the input is structured as $d \times n$ where d is the embedding dimension and n is the sequence length.

We employ the T5-11B architecture from Raffel et al. [4] with **Encoder**_{fusion} and **Decoder**_{Intention}.

Encoder_{fusion} $n_{\text{Enc}} = 24$ with a final dimension $d_T = 1024$. In each layer l , the Multi-Head Self-Attention is made of $H_{\text{Enc}} = 128$ heads and the Feedforward Network has the following composition $((\text{GeGLU}, W_1^{\text{FFN},l}, b_1^{\text{FFN},l}), (\text{Id}, W_2^{\text{FFN},l}, b_2^{\text{FFN},l}))$ with an inner dimension of 65536 i.e. $W_1^{\text{FFN},l} \in \mathbb{R}^{65536 \times 1024}$ and $W_2^{\text{FFN},l} \in \mathbb{R}^{1024 \times 65536}$.

Decoder_{Intention} $n_{\text{Dec}} = 24$ with a final dimension $d_T = 1024$. In each layer l , the Masked Multi-Head Self-Attention is made of $H_{\text{Dec}}^{\text{self}} = 128$ heads, the Multi-Head Cross-Attention is made of $H_{\text{Dec}}^{\text{cross}} = 128$ heads. The Feedforward Network $((\text{GeGLU}, W_1^{\text{FFN},l}, b_1^{\text{FFN},l}), (\text{Id}, W_2^{\text{FFN},l}, b_2^{\text{FFN},l}))$ has an inner dimension of 65536 i.e. $W_1^{\text{FFN},l} \in \mathbb{R}^{65536 \times 1024}$ and $W_2^{\text{FFN},l} \in \mathbb{R}^{1024 \times 65536}$.

B Mathematical Interpretations

B.1 Workflow Signal

B.1.1 Algebraic Foundations of Workflow Signals

Let X be a tensor space over the field of real numbers \mathbb{R} . We assume X is a finite-dimensional real Hilbert space equipped with an inner product $\langle \cdot, \cdot \rangle : X \times X \rightarrow \mathbb{R}$, hence satisfying the following properties for all $x, y, z \in X$ and $\alpha \in \mathbb{R}$:

1. Symmetry: $\langle x, y \rangle = \langle y, x \rangle$
2. Linearity: $\langle \alpha x + y, z \rangle = \alpha \langle x, z \rangle + \langle y, z \rangle$
3. Positive definiteness: $\langle x, x \rangle \geq 0$, with equality if and only if $x = 0$

This inner product induces a norm:

$$\|\cdot\| : \begin{cases} X \rightarrow \mathbb{R} \\ x \mapsto \sqrt{\langle x, x \rangle} \end{cases}$$

Which in turn defines a metric:

$$d : \begin{cases} X \times X \rightarrow \mathbb{R} \\ x, y \mapsto \|x - y\| \end{cases}$$

The completeness of X with respect to this metric is guaranteed by our assumption that X is a Hilbert space, meaning that every Cauchy sequence in X converges to an element in X . Let $\{x_n\}_{n \in \mathbb{N}}$ a Cauchy sequence in X ($\forall \varepsilon > 0, \exists N \in \mathbb{N}$ such that $\forall n, m > N, \|x_n - x_m\| < \varepsilon$), there exists $x \in X$ such that $\lim_{n \rightarrow \infty} \|x_n - x\| = 0$.

Given the finite dimensionality of X , we can construct an orthonormal basis $\{e_i\}_{i=1}^n$ of X , where $n = \dim(X)$, satisfying:

1. Orthonormality: $\forall i, j, \langle e_i, e_j \rangle = \delta_{ij}$ (where δ_{ij} is the Kronecker delta)
2. Completeness: $\text{span}(\{e_i\}_{i=1}^n) = X$

This orthonormal basis allows for the unique representation of any Workflow Signal thread vector $x \in X$ as a finite linear combination of $\{e_i\}_{i=1}^n$:

$$\forall x \in X, \exists! (\alpha_i)_{i=1}^n \text{ such that } x = \sum_{i=1}^n \alpha_i e_i, \text{ where } \alpha_i = \langle x, e_i \rangle$$

The following aims to define a subspace S of X of Workflow Signals.

Let S be a non-empty subspace of $X : S \subseteq X$. S has the following properties:

1. Finite dimensionality: $\dim(S) < \infty$
2. Inner product structure: $\exists \langle \cdot, \cdot \rangle_S : S \times S \rightarrow \mathbb{R}$
3. Completeness: S is complete under $\|\cdot\|_S = \sqrt{\langle \cdot, \cdot \rangle_S}$

Let I, P, O be subspaces of S corresponding to Input, Process and Output Workflow Signals:

$$S = I \oplus P \oplus O \tag{57}$$

where \oplus denotes the direct sum between two spaces, such that:

$$\forall s \in S, \exists! i \in I, p \in P, o \in O, \text{ such that } s = i + p + o$$

and

$$I \cap P = \{0_S\}, O \cap P = \{0_S\} \text{ and } I \cap O = \{0_S\}$$

The assumption of the direct sum decomposition of S into the subspaces I, P , and O originates from the real world business context of Opus. The space X represents the context space, in which a set of Business Artefacts is encoded as one vector. The signal space $S \subseteq X$ consists of Workflow Signals. In this framework, the Input representation of a Workflow Signal is determined by the context that defines it as an Input. While Input and Output may sometimes refer to the same underlying object, their representations remain distinct within a given context. For instance, a “medical record” can function as both an Input and an Output in a Process, but within a Workflow Signal describing a specific Workflow, the representation of “medical record” as an Input will differ from its representation as an Output. This distinction is context-dependent and ensures that Input and Output roles remain well-defined within the Workflow Signal space. Therefore we stipulate that each Workflow Signal can be uniquely decomposed into Input, Process and Output components.

We suppose that I, P, and O are non empty.

As subspaces of S, we can define $\{e_{I,k}\}_{k=1}^{\dim(I)}$, $\{e_{P,k}\}_{k=1}^{\dim(P)}$, $\{e_{O,k}\}_{k=1}^{\dim(O)}$ orthonormal bases of I, P and O respectively.

We can define the projection operators $p_I : S \rightarrow I$, $p_P : S \rightarrow P$ and $p_O : S \rightarrow O$ such that:

$$\begin{aligned} p_I + p_P + p_O &= \text{Id}_S \\ \forall s \in S, (p_I + p_P + p_O)(s) &= s \\ \text{Im}(p_I) + \text{Im}(p_P) + \text{Im}(p_O) &= S \end{aligned}$$

and

$$p_I^2 = p_I, p_P^2 = p_P, p_O^2 = p_O$$

Based on the above, $s \in S$ can be decomposed as:

$$s = p_I(s) + p_P(s) + p_O(s) \quad (58)$$

And each of i , p and o can be uniquely decomposed on their respective bases:

$$s = \sum_{k=1}^{\dim(I)} \alpha_{I,k} e_{I,k} + \sum_{k=1}^{\dim(P)} \alpha_{P,k} e_{P,k} + \sum_{k=1}^{\dim(O)} \alpha_{O,k} e_{O,k} \quad (59)$$

This formalism enables the expression of generative families within the spaces I, P and O, which serve as the foundational idea for the class sets of the classification heads employed in the system.

B.1.2 Generative families of I, P, O

Definition Let I_g, P_g, O_g be generative families of I, P, O respectively.

For $X \in \{I, P, O\}$, $X_g = \{e_{X_g,k}\}_k$, $\forall k, e_{X_g,k} \in X \forall x \in X, \exists (\alpha_{X,k})_k$ such that $x = \sum_{k=1}^{|X_g|} \alpha_{X,k} e_{X_g,k}$

The Input, Process and Output generative families can be built initially semantically using Large Language Models and iteratively updated from Workflow Signals as described in Algorithm 1.

Algorithm 1 Adaptive Construction of Generative Family from Intention with Error Control

Require:

- 1: A Workflow Signal $x \in X, X \in \{I, P, O\}$
- 2: Initial generative family: $X_g = \{e_{X_g, k}\}_{k=1}^{|X_g|} \subset X$,
- 3: Error threshold $\epsilon_X > 0$
- 4: Maximum iteration count M_{\max}

Ensure: Coefficient vectors $\alpha_X \in \mathbb{R}^{|X_g|}$, and sets such that

$$\left\| x - \sum_{k=1}^{|X_g|} \alpha_{X, k} e_{X_g, k} \right\| < \epsilon_X,$$

- 5: **function** DECOMPOSEWITHERROR($x, X_g, \epsilon, M_{max}$)
 - 6: $m \leftarrow 0$
 - 7: $error \leftarrow \infty$
 - 8: **while** $error > \epsilon$ AND $m < M_{max}$ **do**
 - 9: **Solve** $\alpha_X^* \leftarrow \operatorname{argmin}_{\alpha_X} \left\| x - \sum_{k=1}^{|X_g|} \alpha_{X, k} e_{X_g, k} \right\|$
 - 10: $error \leftarrow \left\| x - \sum_{k=1}^{|X_g|} \alpha_{X, k}^* e_{X_g, k} \right\|$
 - 11: **if** $error > \epsilon$ **then**
 - 12: $X_g \leftarrow X_g \cup \{x\}$
 - 13: **end if**
 - 14: $m \leftarrow m + 1$
 - 15: **end while return** α_X^*, X_g
 - 16: **end function**
 - 17: **procedure** MAIN
 - 18: $(\alpha_X, X_g) \leftarrow \text{DECOMPOSEWITHERROR}(x, X_g, \epsilon_X, M_{\max})$
 - 19: **end procedure**
-

Algorithm 1 exhibits potentially large complexity: the worst-case iteration count of M_{\max} could be reached for each component, yielding $\mathcal{O}(3M_{\max})$ iterations with each iteration solving an increasingly complex minimization problem. However, as the Input, Process and Output spaces are relatively constrained, a solution is reached within a bounded number of iterations and acceptable computational complexity.

Our system’s parameter complexity and training cost are directly influenced by the size and stability of the generative families for Input, Process and Output classifications. Therefore, Algorithm 1 is systematically combined with a Gram–Schmidt-type algorithm to control the dimensionality of these families, which naturally expand when using Algorithm 1 alone. This process ensures convergence to a stable generative family structure. In practice, as we construct these generative families from granular signals i , o and p (extracted from Business Artefacts), we enforce a rigorous dimensionality control mechanism—refining these families as new elements are incorporated.

B.2 Workflow Intentions

B.2.1 Algebraic Foundations of Workflow Intention

Let $\mathcal{G} = \text{I} \times \text{P} \times \text{O}$, $\gamma \in \mathcal{G}$ is an ordered triple representing a Workflow Intention in terms of Input, Output and Process Workflow Signals.

$$\dim(\mathcal{G}) = \dim(\text{I}) + \dim(\text{P}) + \dim(\text{O})$$

The canonical projections on \mathcal{G} are

$$\begin{aligned} \pi_{\text{I}} : \mathcal{G} &\rightarrow \text{I}, & \forall \gamma = (i, p, o) \in \mathcal{G} : \pi_{\text{I}}(\gamma) &= i \\ \pi_{\text{P}} : \mathcal{G} &\rightarrow \text{P}, & \forall \gamma = (i, p, o) \in \mathcal{G} : \pi_{\text{P}}(\gamma) &= p \\ \pi_{\text{O}} : \mathcal{G} &\rightarrow \text{O}, & \forall \gamma = (i, p, o) \in \mathcal{G} : \pi_{\text{O}}(\gamma) &= o \end{aligned} \quad (60)$$

With kernels in \mathcal{G} :

$$\begin{aligned} \ker(\pi_{\text{I}}) &= \{\gamma \in \mathcal{G} : \gamma = (0_{\text{I}}, o, p)\} \\ \ker(\pi_{\text{P}}) &= \{\gamma \in \mathcal{G} : \gamma = (i, 0_{\text{P}}, o)\} \\ \ker(\pi_{\text{O}}) &= \{\gamma \in \mathcal{G} : \gamma = (i, p, 0_{\text{O}})\} \end{aligned} \quad (61)$$

Finally, for a topology $\tau_{\mathcal{G}}$, $\forall U \in \tau_{\mathcal{G}} : U \subseteq \mathcal{G}$ and $\forall \gamma \in U$, there exist open neighborhoods $B_{\text{I}}(\gamma) \in \tau_{\text{I}}$, $B_{\text{P}}(\gamma) \in \tau_{\text{P}}$, and $B_{\text{O}}(\gamma) \in \tau_{\text{O}}$ such that $\gamma \in (B_{\text{I}}(\gamma) \times B_{\text{P}}(\gamma) \times B_{\text{O}}(\gamma)) \subseteq U$. This allows the Opus system to understand “closeness” of Workflow Intentions, that is, if something is “close” in \mathcal{G} , it is close in all three components (I, P, and O) simultaneously.

A single Workflow Intention γ in \mathcal{G} is defined as:

$$\gamma = (i, p, o) \in \mathcal{G} \text{ such that } i \in \text{I}, p \in \text{P} \text{ and } o \in \text{O} \quad (62)$$

Workflow Intentions can be combined to accommodate users with hybrid Workflow Intention:

$$\gamma_1 + \gamma_2 = (i_1, p_1, o_1) + (i_2, p_2, o_2) = (i_1 + i_2, p_1 + p_2, o_1 + o_2) \quad (63)$$

Workflow Intention representations can have different strengths i.e. expressiveness in different contexts

$$\alpha\gamma = \alpha(i, p, o) = (\alpha i, \alpha p, \alpha o) \quad (64)$$

B.2.2 Workflow Intentions from Workflow Signal

Let $f : \text{X} \times \text{S} \mapsto \mathcal{P}(\mathcal{G})$ the powerset of \mathcal{G} . By definition, $\mathcal{P}(\mathcal{G}) = \{A \mid \forall x \in A, x \in \mathcal{G}\}$.

Let $(x, s) \in \text{X} \times \text{S}$,

$$\begin{aligned} \exists n \in \mathbb{N}, \Gamma = \{\gamma_k\}_{k=1}^n \text{ such that } f(x, s) = \Gamma \text{ and} \\ \forall \gamma \in \Gamma, \gamma \in \mathcal{G} \text{ and } \exists i_\gamma \in \text{I}, p_\gamma \in \text{P}, o_\gamma \in \text{O} \text{ such that } \gamma = (i_\gamma, p_\gamma, o_\gamma) \end{aligned} \quad (65)$$

Property: Information Conservation Let $x \in X, s \in S$ and $\Gamma = f(x, s)$.

$$\forall \gamma \text{ in } f(x, s), \quad \|i_\gamma\| \leq \|i_s\|, \|p_\gamma\| \leq \|p_s\| \text{ and } \|o_\gamma\| \leq \|o_s\| \quad (66)$$

Any Workflow Signal (Input, Process or Output) of a Workflow Intention object of a Workflow Intention Set is weaker or equal than the overall Workflow Signal (Input, Process or Output) the Workflow Intention Set was derived from.

Property: Workflow Intention Variation Let $x \in X, s \in S$ and $\Gamma = f(x, s)$, if $|f(x, s)| > 1$,

$$\forall \gamma_1, \gamma_2 \in f(x, s), \left| \left\{ v, \frac{\langle v_{\gamma_1}, v_{\gamma_2} \rangle}{\|v_{\gamma_1}\| \|v_{\gamma_2}\|} < \epsilon_{\text{sim}}, v \in \{i, p, o\} \right\} \right| \geq 1, \quad \epsilon_{\text{sim}} \in [0, 1[\quad (67)$$

$$\forall \gamma_1, \gamma_2 \in f(x, s), \left| \left\{ v, \|v_{\gamma_1} - v_{\gamma_2}\| > \epsilon_2, v \in \{i, p, o\} \right\} \right| \geq 1, \quad \epsilon_2 \in [0, 1[\quad (68)$$

If the Workflow Intention Set is composed of two or more Workflow Intention objects, each pair of Workflow Intention must present variation on at least one Workflow Signal dimension (Input, Process or Output).