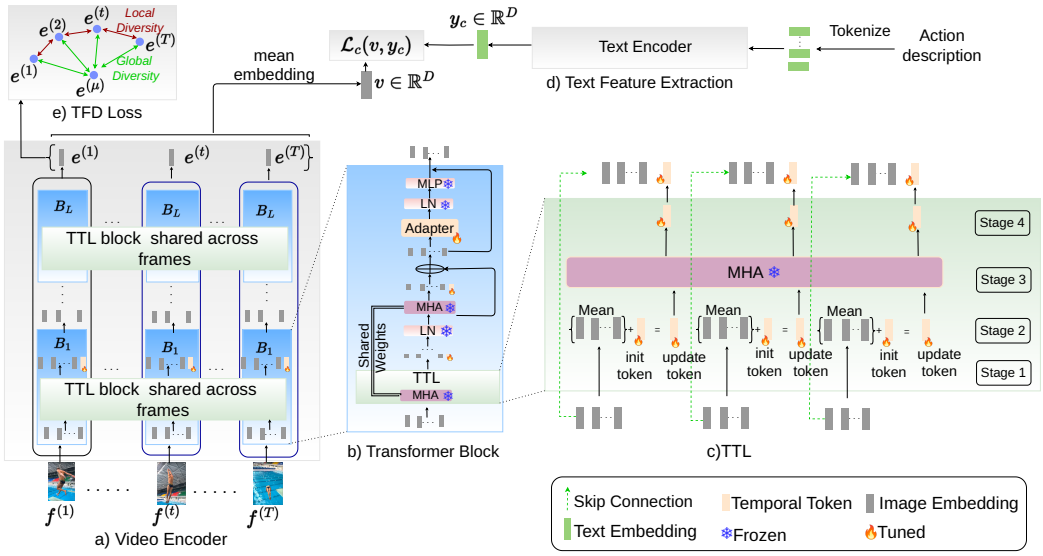

Figure 1: ***Overview of the proposed method:*** T2L utilizes Temporal Token Learning (TTL) and Temporal Feature Diversity Loss (TFD) to efficiently learn temporal aspects, eliminating the need for the frame integration module, a bottleneck in adapting image models for video understanding. Module (a) depicts the architecture of a video encoder, where each frame is parallel encoded by a vision transformer. Within the vision transformer, each block is adapted using spatial adapters shown in (b). Temporal relations between frames are learned by module (c) - Temporal Token Learning. Module (d) shows the text encoder, and module (e) shows the Temporal Feature Diversity Loss $\mathcal{L}_{TFD}$).

0.0.1 Temporal Token Learning

To facilitate cross-frame temporal learning in a video, we introduce a novel concept called Temporal Token Learning. The following steps outline the key components and process:

1. **Incorporation of Temporal Tokens:**
   - Temporal tokens are strategically incorporated at the input space of each Transformer layer.
   - Denoted as $P^{Temp} \in \mathbb{R}^{L \times T \times D}$, where L is the number of layers in the CLIP Transformer.
   - The collection of input learnable tokens is defined as $P^{Temp} = \{p_l \in \mathbb{R}^{T \times D} \mid l = 0, \ldots, L-1\}$.
2. **Linking Temporal Tokens with Frame Embeddings:**
   - Each t-th frame's embedding is linked to the respective temporal token as:

$$\tilde{p}_l^{(t)} = p_l^{(t)} + \frac{1}{N+1} \sum_{j=1}^{N+1} (z_{l-1}^{(t)})_j. \tag{1}$$

   - This process is illustrated in Figure 1 module (d).
3. **Duplication of Multi-Head Attention (MHA) Block:**
   - To capture temporal dependencies effectively, we duplicate the MHA block within each transformer layer.
   - The duplicated MHA block is initialized using the same parameters as the existing MHA layer, ensuring consistency and reducing computational complexity.
   - This new MHA block is dedicated to processing only the temporal tokens.
4. **Processing Temporal Tokens:**
   - Temporal tokens are obtained by aggregating the average pooling of each frame token and the learnable embeddings.

- Subsequently, MHA is performed on all tokens for each frame independently (including the temporal token). This operation at the l-th block is expressed as:

$$\hat{p}_l = \text{MHA}(\text{LN}(\tilde{p}_l)). \tag{2}$$

- Here, $p_l = [p_l^{(1)}, p_l^{(2)}, \ldots, p_l^{(T)}]$.
- MHA represents a pre-trained and frozen attention block at the l-th layer derived from CLIP, as shown in Figure 1 module (c).
- LN denotes layer normalization.

5. **Final Concatenation:**
   - The final learned temporal token is concatenated with the frame embedding $[z_{l-1}^{(t)}, t_l^{(t)}]$ to facilitate further processing.
6. **Novelty and Efficiency:**
   - Our Temporal Token Learning approach is novel and unique as it enables efficient temporal learning without a separate architecture.
   - By leveraging the existing CLIP image MHA with self-learnable tokens, we simplify the design while effectively capturing temporal dependencies.
   - This makes it a highly efficient solution for temporal learning in video processing.