The paper "DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification" introduces several key formulas to achieve dynamic token sparsification. Here are the main formulas used in the paper:

1. **Local Feature Projection** (Equation 1):

$$
z_{\text{local}} = \text{MLP}(x) \in \mathbb{R}^{N \times C'}
$$

- Projects tokens x to a lower dimension C' (typically $C' = C/2$) using an MLP.

2. **Global Feature Aggregation** (Equation 2):

$$
z_{\text{global}} = \text{Agg}(\text{MLP}(x), \hat{D}) \in \mathbb{R}^{C'}
$$

- Aggregates global context from tokens using a decision mask $\hat{D}$. The aggregation function Agg is typically average pooling (Equation 3):

$$
\text{Agg}(u, \hat{D}) = \frac{\sum_{i=1}^N \hat{D}_i u_i}{\sum_{i=1}^N \hat{D}_i}, \quad u \in \mathbb{R}^{N \times C'}
$$

3. **Local-Global Embedding** (Equation 4):

$$
z_i = [z_{\text{local}_i}, z_{\text{global}_i}], \quad 1 \leq i \leq N
$$

- Combines local and global features for each token.

4. **Token Retention Probability** (Equation 5):

$$
\pi = \text{Softmax}(\text{MLP}(z)) \in \mathbb{R}^{N \times 2}
$$

- Predicts probabilities $\pi_{i,0}$ (drop) and $\pi_{i,1}$ (keep) for each token.

5. **Gumbel-Softmax Sampling** (Equation 7):

$$
D = \text{Gumbel-Softmax}(\pi)_{*,1} \in \{0,1\}^N
$$

- Samples binary decisions D (0: drop, 1: keep) differentiably.

6. **Attention Masking** (Equations 9–11):

- Computes attention scores $P = QK^T / \sqrt{C}$.
- Constructs a masking graph G (Equation 10):

G_{ij} = \begin{cases}
1 & \text{if } i=j, \\

\hat{D}_j & \text{if } i \neq j.
\end{cases}
\]
- Applies masking to the attention matrix \(\tilde{A} \) (Equation 11):
\[
\tilde{A}_{ij} = \frac{\exp(P_{ij}) G_{ij}}{\sum_{k=1}^N \exp(P_{ik}) G_{ik}}
\]

7. **Training Objectives**:
- Cross-entropy loss \(L_{\text{cls}} \).
- Self-distillation loss \(L_{\text{distill}} \) (Equation 13).
- KL divergence loss \(L_{\text{KL}} \) (Equation 14).
- Token ratio loss \(L_{\text{ratio}} \) (Equation 15).

These formulas enable hierarchical token pruning, differentiable training, and efficient inference. For details, refer to the paper's Sections 3.2–3.4.