Self-Paced and Self-Corrective Masked Prediction for Movie Trailer Generation

Supplementary Material

6. Model Architecture

Our SSMP model adopts a Transformer-based encoder architecture that leverages full self-attention across all shots. Given an input sequence of shot features $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_T]$ as the initial hidden state $\mathbf{h}^{(0)}$, the model processes it through a stack of L Transformer encoder blocks. Each block consists of a full self-attention layer and a feed-forward layer, both surrounded by residual connections and normalized using RMSNorm.

At the l-th layer, the hidden representation $\mathbf{h}^{(l)} \in \mathcal{R}^{T\times D}$ is updated as:

$$\begin{aligned} \mathbf{h}'^{(l)} &= \mathbf{h}^{(l)} + \mathrm{MSA}\big(\mathrm{RMSNorm}(\mathbf{h}^{(l)})\big), \\ \mathbf{h}^{(l+1)} &= \mathbf{h}'^{(l)} + \mathrm{FFN}\big(\mathrm{RMSNorm}(\mathbf{h}'^{(l)})\big), \end{aligned} \tag{9}$$

where $\mathrm{MSA}(\cdot)$ denotes a multi-head self-attention mechanism formulated as:

$$\begin{aligned} \mathbf{Q} = \mathbf{h}^{(l)}W_Q, \quad \mathbf{K} = \mathbf{h}^{(l)}W_K, \quad \mathbf{V} = \mathbf{h}^{(l)}W_V, \\ \mathrm{MSA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}\left(\frac{(\mathbf{Q}R_\theta)(\mathbf{K}R_\theta)^\top}{\sqrt{D_k}}\right)\mathbf{V}, \end{aligned} \tag{10}$$

where R_θ represents the rotary positional embedding matrix [42] to encode relative positions, and W_Q, W_K and W_V are learnable matrices. The feed-forward network (FFN) takes $\mathbf{z} = \mathrm{RMSNorm}(\mathbf{h}'^{(l)})$ as input and applies a gated SiLU activation with a linear expansion and projection:

$$\mathrm{FFN}(\mathbf{z}) = \mathrm{Linear}_{\mathrm{out}}(\mathrm{SiLU}(\mathrm{Linear}_{\mathrm{up}}(\mathbf{z}))), \tag{11}$$

where $\mathrm{Linear}_{\mathrm{up}}$: $\mathbb{R}^D \to \mathbb{R}^{2D}$, and $\mathrm{Linear}_{\mathrm{out}}$: $\mathbb{R}^{2D} \to \mathbb{R}^D$. Referring to [33], our model replaces standard layer normalization with RMS normalization:

$$\mathrm{RMSNorm}(\mathbf{h}) = \frac{\mathbf{h}}{\sqrt{\frac{1}{D}\|\mathbf{h}\|_2^2 + \epsilon}} \odot \mathbf{g}, \tag{12}$$

where $\mathbf{g}$ is a learnable scaling vector and ϵ is a small constant ensuring numerical stability. This encoder architecture allows each shot to attend to all others, enabling bidirectional context modeling.

7. Post-Processing Details

As mentioned in Sec. 3.4, we insert the selected narrations into the generated trailer through solving a dynamic programming problem. Formally, given the similarity matrix $\boldsymbol{C} = [c_{n,j}] \in \mathbb{R}^{N\times J}$ between N narration features and J trailer shot description features, the optimal alignment is obtained by maximizing the cumulative similarity under the constraint that each narration audio duration L_j^{shot} must not exceed the duration of its corresponding trailer shot L_n^{nar}, *i.e.*, if $L_j^{\mathrm{shot}} \geq L_n^{\mathrm{nar}}$,

$$D_{n,j} = \max\{D_{n-1,\,j-1} + c_{n,j},\ D_{n,\,j-1}\}, \tag{13}$$

where $D_{n,j}$ denotes the maximum accumulated similarity between the first n narrations and the first j trailer shots. We initialize $D_{0,*} = 0$ and $D_{*,0} = -\infty$. The optimal alignment path is then obtained by tracing back from $\arg\max_j D_{N,j}$. After inserting the background music and the narrations into the trailer $\mathcal{T}_{\boldsymbol{z}}$, we obtain the final generated trailer with coherent audiovisual composition.

8. Detailed settings of the user study.

We conduct the user study with 25 participants aged between 10 and 52. Each participant evaluate trailers for three movies used in [50, 51] (*The Hobbit 2*, *The Wolverine*, and *300: Rise of an Empire*). For each movie, participants watch and rate official trailer and six anonymized trailers generated by our method and five baselines.

9. More Experimental Results

Mask Ratio Momentum. To further illustrate the results in Tab. 4, Fig. 5 presents how the masking ratio changes during the middle of the training. The curve of $\mu_t = 0.1$ lies between those of 0.0, 0.5, and 0.9, suggesting a balanced update of the mask ratio, achieving a trade-off between training efficiency and stability to accuracy changes.

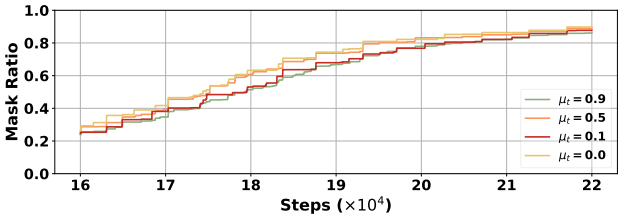

Figure 5. Mask ratio over training steps for different settings.

Training Accuracy Momentum. Tab. 8 shows the results with different values of the hyperparameter μ_a, which controls the balance between the current accuracy and its accumulated momentum during training. The best results occur at $\mu_a = 0.98$, and the overall stability across different values further demonstrates its robustness.

Visual Encoder. Since previous trailer generation methods [50, 54] adopt ImageBind, we follow the same setting for a fair comparison and demonstrate that the performance gain is attributed by our method rather than by the

Table 8. The robustness test of accuracy momentum.

Accuracy Momentum	Test-ALL				
	Precision↑	Recall↑	F1↑	LD↓	AA↑
$\mu_a = 1.0$	0.1661	0.1925	0.1780	93.06	0.67
$\mu_a = 0.98$(our)	**0.1888**	**0.2126**	**0.1996**	**91.19**	**0.68**
$\mu_a = 0.95$	0.1743	0.1888	0.1854	91.98	**0.68**
$\mu_a = 0.9$	0.1614	0.1872	0.1731	92.90	0.64
$\mu_a = 0.5$	0.1636	0.1894	0.1753	93.34	0.66
$\mu_a = 0.0$	0.1585	0.1836	0.1698	93.34	0.65

encoder. Secondly, unlike image-based models (DINOv3 and SigLIP2), ImageBind provides a native visual encoder capturing spatio-temporal dynamics of shot, not requiring additional pooling of image features. We replace ImageBind with SigLIP2 and DINOv3, learning shot-level representations under the same experimental setup. Tab. 9 shows that the three encoders are comparable in shot selection, verifying the rationality of using ImageBind.

Table 9. Testing different encoders on Test-ALL for shot selection.

Visual Encoder	Precision↑	Recall↑	F1↑	LD↓	AA↑
SigLIP2	0.1873	0.2018	0.1936	93.38	0.67
DINOv3	0.1894	0.2106	0.1990	93.33	0.66
ImageBind (Ours)	0.1888	0.2126	0.1996	91.19	0.68

Self-correction Mechanism. Fig. 6 offers some insights into the self-correction across positions. i) The remasking and revision counts are unevenly distributed across positions, indicating that self-correction is selectively applied. ii) Most positions are re-masked at early stages. As some positions become fixed, re-masking gradually concentrates on some key positions. iii) The revision count is much lower than the remasking count — remasking leads to a revision only when sufficient confidence is reached. iv) *Notably, when applying a relaxed shot selection metric (e.g., allowable positional deviation=3), the superiority of self-correction becomes significant, as shown in Tab. 10.*

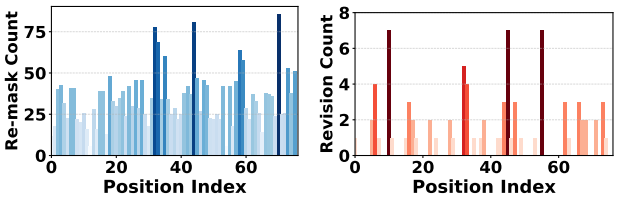

Figure 6. Remasking and revision counts at each position index in a single trailer of 76 shots during the generation process.

Bi-directional Modeling. Notably, our self-paced training and self-correction mechanisms are specifically designed for bi-directional models and cannot be directly applied to autoregressive models. For a fair comparison, we align our model architecture (e.g., input and hidden dimensions, etc.) with the autoregressive method TGT [1] and compare them in Tab. 11. The bi-directional model learned by our method outperforms autoregression even using fewer layers.

[1]SigLIP 2 adopts `siglip2-large-patch16-256`.
[2]DINOv3 adopts `dinov3-vitl16-pretrain-lvd1689m`.

Table 10. Numerical results under a relaxed shot selection metric.

Strategy	Test-8			Test-74		
	Precision↑	Recall↑	F1↑	Precision↑	Recall↑	F1↑
Greedy	0.533	0.584	0.557	0.535	0.560	0.544
Self-corrective	**0.676**	**0.619**	**0.631**	**0.582**	**0.670**	**0.622**

Table 11. Bi-directional model (ours) v.s. Autoregression (TGT).

Method	Test-ALL				
	Precision↑	Recall↑	F1↑	LD↓	AA↑
TGT [1] 5-layer	0.1214	0.1288	0.1240	96.08	0.46
SSMP 4-layer	0.1492	0.1734	0.1601	88.51	0.66
SSMP 5-layer	**0.1596**	**0.1847**	**0.1710**	**87.93**	**0.68**
SSMP 8-layer	0.1577	0.1836	0.1694	88.09	**0.68**

Model architecture. As shown in Tab. 11, the 5-layer model achieves the best results. The 4-layer and 8-layer models suffer from under- and over-fitting, respectively. We adopt the 5-layer model, considering the current limited data scale. In the future, we will expand the dataset for training a larger model.

Movie Time Periods and Genres. Tab. 12 shows that our method performs best for movies between 2010 and 2020, likely because most of the training data comes from this period. In addition, our method achieves better performance on Crime, Thriller, and Comedy movies, while the performance on Adventure movies is relatively weaker. This may be attributed to the higher variability in visual content and narrative structure in Adventure films, as well as the potential genre imbalance in the training data.

Table 12. Results on movies across time periods and genres.

Time Periods Genres	#Movies	Test-ALL				
		Precision↑	Recall↑	F1↑	LD↓	AA↑
2000-2010	25	0.1726	0.1971	0.1838	93.86	0.68
2010-2020	57	**0.1920**	**0.2220**	**0.2056**	**88.69**	**0.69**
after 2020	30	0.1587	0.1999	0.1759	93.29	0.60
Thriller	43	0.2031	0.2259	0.2132	90.56	**0.68**
Action	30	0.1964	0.2172	0.2060	93.25	0.65
Comedy	30	0.2104	0.2400	0.2240	91.75	0.62
Adventure	23	0.1742	0.2041	0.1877	83.56	**0.68**
Crime	23	**0.2311**	**0.2543**	**0.2419**	**82.53**	0.66