

Appendix

Cross-Mamba computation

We formalize the computational framework of Cross-Mamba, which enables synergistic fusion of spatial and spectral features through dual-path state-space modeling. Let $\mathbf{F}_{\text{sp}} \in \mathbb{R}^{L_s \times d}$ and $\mathbf{F}_{\text{freq}} \in \mathbb{R}^{L_f \times d}$ denote spatial and spectral feature sequences, respectively. The dual-path SSM transformations are governed by:

$$\begin{aligned} \text{Path A: } \mathbf{Y}_a &= \text{SSM}_{\Theta_a}(\mathbf{X}_{\text{freq}}), \\ \Theta_a &= \{\mathbf{A}, \Delta, \mathbf{B}_{\text{freq}}, \mathbf{C}_{\text{sp}}\}, \\ \text{Path B: } \mathbf{Y}_b &= \text{SSM}_{\Theta_b}(\mathbf{X}_{\text{sp}}), \\ \Theta_b &= \{\mathbf{A}, \Delta, \mathbf{B}_{\text{sp}}, \mathbf{C}_{\text{freq}}\}. \end{aligned} \quad (15)$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ and $\Delta \in \mathbb{R}$ are shared parameters maintaining consistent state dynamics. The cross-modal projections are defined as:

$$\begin{aligned} \mathbf{C}_{\text{sp}} &= \text{Linear}_c(\mathbf{F}_{\text{sp}}), \\ \mathbf{B}_{\text{freq}}, \mathbf{X}_{\text{freq}} &= \text{Linear}_{bx}(\mathbf{F}_{\text{freq}}), \\ \mathbf{C}_{\text{freq}} &= \text{Linear}_c(\mathbf{F}_{\text{freq}}), \\ \mathbf{B}_{\text{sp}}, \mathbf{X}_{\text{sp}} &= \text{Linear}_{bx}(\mathbf{F}_{\text{sp}}). \end{aligned} \quad (16)$$

Discretizing the continuous SSM via zero-order hold yields:

$$\begin{aligned} \bar{\mathbf{A}} &= \exp(\Delta \mathbf{A}), \\ \bar{\mathbf{B}}_{\text{freq}} &= (\Delta \mathbf{A})^{-1} (e^{\Delta \mathbf{A}} - \mathbf{I}) \mathbf{B}_{\text{freq}}, \\ \bar{\mathbf{B}}_{\text{sp}} &= (\Delta \mathbf{A})^{-1} (e^{\Delta \mathbf{A}} - \mathbf{I}) \mathbf{B}_{\text{sp}}. \end{aligned} \quad (17)$$

The recurrent forms for time step k are:

$$\begin{aligned} h_k^{(a)} &= \bar{\mathbf{A}} h_{k-1}^{(a)} + \bar{\mathbf{B}}_{\text{freq}} x_k^{(\text{freq})}, \\ y_k^{(a)} &= \mathbf{C}_{\text{sp}} h_k^{(a)}, \\ h_k^{(b)} &= \bar{\mathbf{A}} h_{k-1}^{(b)} + \bar{\mathbf{B}}_{\text{sp}} x_k^{(\text{sp})}, \\ y_k^{(b)} &= \mathbf{C}_{\text{freq}} h_k^{(b)}, \end{aligned} \quad (18)$$

Expanding the recurrence reveals the attention-like formulation:

$$\begin{aligned} \mathbf{Y}_a &= \mathbf{C}_{\text{sp}} \sum_{j=1}^{L_f} \left(\prod_{k=j+1}^{L_f} \bar{\mathbf{A}}_k \right) \bar{\mathbf{B}}_{\text{freq},j} \mathbf{X}_{\text{freq},j}, \\ \mathbf{Y}_b &= \mathbf{C}_{\text{freq}} \sum_{j=1}^{L_s} \left(\prod_{k=j+1}^{L_s} \bar{\mathbf{A}}_k \right) \bar{\mathbf{B}}_{\text{sp},j} \mathbf{X}_{\text{sp},j}. \end{aligned} \quad (19)$$

Applying the exponential property $\prod \exp(\mathbf{M}_k) = \exp(\sum \mathbf{M}_k)$:

$$\begin{aligned} \mathbf{Y}_a &= \mathbf{C}_{\text{sp}} \sum_{j=1}^{L_f} \exp \left(\sum_{k=j+1}^{L_f} \Delta_k \mathbf{A} \right) \bar{\mathbf{B}}_{\text{freq},j} \mathbf{X}_{\text{freq},j}, \\ \mathbf{Y}_b &= \mathbf{C}_{\text{freq}} \sum_{j=1}^{L_s} \exp \left(\sum_{k=j+1}^{L_s} \Delta_k \mathbf{A} \right) \bar{\mathbf{B}}_{\text{sp},j} \mathbf{X}_{\text{sp},j}. \end{aligned} \quad (20)$$

The final fusion integrates both paths through linear projection:

$$\begin{aligned} \mathbf{Y} &= \text{Linear}_{\text{fuse}}([\mathbf{Y}_a; \mathbf{Y}_b]), \\ &= \mathbf{W}_f [\mathbf{Y}_a \parallel \mathbf{Y}_b] + \mathbf{b}_f. \end{aligned} \quad (21)$$

This establishes bidirectional co-modulation: Spatial features \mathbf{F}_{sp} guide spectral synthesis via \mathbf{C}_{sp} in Path A, while spectral features \mathbf{F}_{freq} regularize spatial generation via \mathbf{C}_{freq} in Path B. The shared state transition matrix \mathbf{A} ensures coherent integration of complementary representations while maintaining linear complexity $\mathcal{O}(L)$.

Structure-Aware Transfer Computation

To better understand how our Cross-Mamba module propagates structural priors from text to visual features, we formulate its computation as a structure-aware transformation process. This builds on a state-space formulation adapted from (Ali, Zimerman, and Wolf 2025), where SSM dynamics simulate attention-like interactions via parameterized recurrence.

We start from the standard SSM update:

$$\begin{aligned} \mathbf{h}_k &= \bar{\mathbf{A}}_k \mathbf{h}_{k-1} + \bar{\mathbf{B}}_k \mathbf{x}_k, \\ \mathbf{y}_k &= \mathbf{C}_k \mathbf{h}_k. \end{aligned} \quad (22)$$

To integrate structure from the semantic anchor \mathbf{A}_{prim} , we map the intermediate parameters as:

$$\begin{aligned} \mathbf{C}_i &= W_C(\mathbf{x}_i), \\ \Delta_k &= \text{ReLU}(W_\Delta(\mathbf{x}_k)), \\ \bar{\mathbf{A}}_k &= \exp(\Delta_k \cdot \mathbf{A}), \\ \bar{\mathbf{B}}_j &= \Delta_j \cdot W_B(\mathbf{x}_j). \end{aligned} \quad (23)$$

We then rewrite the output at timestep i as the accumulated contribution from all previous positions:

$$\mathbf{y}_i = \sum_{j=1}^i \mathbf{C}_i \left(\prod_{k=j+1}^i \bar{\mathbf{A}}_k \right) \bar{\mathbf{B}}_j \mathbf{x}_j, \quad (24)$$

This can be interpreted as directional modulation, where the structural role of each \mathbf{x}_j is gated by $\bar{\mathbf{B}}_j$ and modulated by state recurrence $\bar{\mathbf{A}}_k$. The final update mimics cross-attention:

$$\mathbf{y}_i = \mathbf{Q}_i \cdot \mathbf{H}_{i,j} \cdot \mathbf{K}_j \cdot \mathbf{x}_j, \quad (25)$$

where:

$$\begin{aligned} \mathbf{Q}_i &= W_C(\mathbf{x}_i), \\ \mathbf{K}_j &= \text{ReLU}(W_\Delta(\mathbf{x}_j) \cdot W_B(\mathbf{x}_j)), \\ \mathbf{H}_{i,j} &= \exp \left(\sum_{k=j+1}^i W_\Delta(\mathbf{x}_k) \cdot \mathbf{A} \right). \end{aligned} \quad (26)$$

This shows that the output \mathbf{y}_i can be seen as modulated retrieval over prior positions, where structural priors are encoded in \mathbf{K}_j , and state propagation is captured by $\mathbf{H}_{i,j}$. Importantly, \mathbf{Q}_i determines how content (i.e., current spatial features) interacts with these priors.

Algorithm 1: SP-DSS Fusion with Quadrangular Scanning

Input: Text features \mathbf{Z}_t , Visual features \mathbf{Z}_v
Parameter: Specific affine parameters $\gamma_t, \beta_t, \gamma_v, \beta_v$
Output: Enhanced features $\mathbf{O}'_t, \mathbf{O}'_v$

- 1: $\mathbf{Z}_{\text{fused}} \leftarrow \text{Concat}[\mathbf{Z}_t; \mathbf{Z}_v]$ {Sequence length $L = L_t + L_v$ }
- 2: $\mathbf{A}, \Delta \leftarrow \text{Linear}_{\text{shared}}(\mathbf{Z}_{\text{fused}})$ {Shared dynamics parameters}
- 3: $\mathbf{Y} \leftarrow \mathbf{0}$ {Initialize output accumulator}
- 4: **for** $i = 1$ **to** 4 **do**
- 5: $\mathbf{B}_t^{(i)}, \mathbf{C}_t^{(i)}, \mathbf{X}_t^{(i)} \leftarrow \text{Linear}_{\text{dir}, i}(\mathbf{Z}_t)$
- 6: $\mathbf{B}_v^{(i)}, \mathbf{C}_v^{(i)}, \mathbf{X}_v^{(i)} \leftarrow \text{Linear}_{\text{dir}, i}(\mathbf{Z}_v)$
- 7: $\mathbf{B}^{(i)} \leftarrow \text{Concat}[\mathbf{B}_t^{(i)}, \mathbf{B}_v^{(i)}]$
- 8: $\mathbf{C}^{(i)} \leftarrow \text{Concat}[\mathbf{C}_t^{(i)}, \mathbf{C}_v^{(i)}]$
- 9: $\mathbf{X}^{(i)} \leftarrow \text{Concat}[\mathbf{X}_t^{(i)}, \mathbf{X}_v^{(i)}]$
- 10: $\mathbf{Y}^{(i)} \leftarrow \text{SSM}_{\theta_i}(\mathbf{X}^{(i)})$ { $\theta_i = \{\mathbf{A}, \Delta, \mathbf{B}^{(i)}, \mathbf{C}^{(i)}\}$ }
- 11: $\mathbf{Y} \leftarrow \mathbf{Y} + \mathcal{R}_i^{-1}(\mathbf{Y}^{(i)})$ {Spatial restoration and aggregation}
- 12: **end for**
- 13: $\mathbf{Y} \leftarrow \text{LayerNorm}(\mathbf{Y})$
- 14: $\mathbf{Y} \leftarrow \text{Linear}_{\text{proj}}(\mathbf{Y})$
- 15: $\mathbf{O}_t, \mathbf{O}_v \leftarrow \text{Split}(\mathbf{Y}, [L_t, L_v])$
- 16: $\mathbf{O}'_t \leftarrow \sigma(\gamma_t) \odot \mathbf{O}_t + \beta_t$ {Text modulation}
- 17: $\mathbf{O}'_v \leftarrow \sigma(\gamma_v) \odot \mathbf{O}_v + \beta_v$ {Visual modulation}
- 18: **return** $\mathbf{O}'_t, \mathbf{O}'_v$

By making \mathbf{B}, Δ dependent on structure ($\mathbf{x}_j = \mathbf{F}_{\text{down}}$) and \mathbf{C} on content ($\mathbf{x}_i = \mathbf{E}_v$), the SSM transition simulates a structure-aware transformation that aligns visual layout with semantic references. This explains how our Cross-Mamba module transfers reference-aligned priors while preserving directional modulation and spatial coherence.

SP-DSS Fusion modal code

The SP-DSS Fusion module integrates textual and visual information through a multi-directional state-space framework, as outlined in Algorithm 1. It first concatenates the two feature streams and applies a shared linear layer to derive global dynamic parameters (\mathbf{A}, Δ) that summarize common temporal or spatial dependencies across modalities. The model then launches four directional scans; in each scan i , separate projections create modality-specific parameters ($\mathbf{B}^{(i)}, \mathbf{C}^{(i)}, \mathbf{X}^{(i)}$), which are fed into a structured state-space model SSM_{θ_i} . These directional outputs are spatially restored via \mathcal{R}_i^{-1} and summed, ensuring that both local and long-range cues are captured from every orientation. The aggregated representation is subsequently layer-normalized and linearly projected to harmonize scale and dimensionality before being split back into text and vision sequences. Finally, gated modulation with learnable pairs (γ_t, β_t) and (γ_v, β_v) amplifies salient patterns in each modality, yielding bidirectionally grounded features while maintaining an overall linear time complexity $\mathcal{O}(L)$.

Algorithm 2: Image Processing via DC-AE

Input: Image \mathbf{I}
Model: Pretrained Autoencoder DC-AE
Output: Reconstructed image $\hat{\mathbf{I}}$

- 1: $\text{DC-AE} \leftarrow \text{LoadModel}(\text{"XXX"})$
- 2: $\mathbf{I}_{\text{tensor}} \leftarrow \text{Normalize}(\text{ToTensor}(\mathbf{I}), \mu=0.5, \sigma=0.5)$
- 3: $\mathbf{I}_{\text{tensor}} \leftarrow \mathbf{I}_{\text{tensor}}[\text{None}]$ {Add batch dimension}
- 4: $\mathbf{I}_{\text{tensor}} \leftarrow \mathbf{I}_{\text{tensor}}.\text{to}(\text{cuda})$
- 5: $\mathbf{z} \leftarrow \text{DC-AE}.\text{encode}(\mathbf{I}_{\text{tensor}}).\text{latent}$ {Encode image to latent}
- 6: $\hat{\mathbf{I}}_{\text{tensor}} \leftarrow \text{DC-AE}.\text{decode}(\mathbf{z}).\text{sample}$ {Decode latent to reconstructed image}
- 7: $\hat{\mathbf{I}}_{\text{tensor}} \leftarrow \hat{\mathbf{I}}_{\text{tensor}} \times 0.5 + 0.5$ {Denormalize to $[0,1]$ }
- 8: $\text{SaveImage}(\hat{\mathbf{I}}_{\text{tensor}}, \text{"demo_dc_ae.png"})$
- 9: **return** $\hat{\mathbf{I}}$



Figure 8: Uncurated generated samples of CIFAR-10.

DC-AE Inference Pipeline

We use a pre-trained DC-AE (Diffusion-Compatible AutoEncoder) model from the Diffusers library to compress and reconstruct images in Algorithm 2. The input image is first normalized and converted to a tensor. It is then encoded into a latent representation using the encoder module of DC-AE. The latent vector is decoded back into image space by the decoder. Finally, we denormalize the output and save the reconstructed image for visualization. All operations are performed on GPU for efficiency.

More qualitative examples

We present our uncured generated samples of Cifar-10 in Figure 8, MS-COCO in Figure 9, MM-CelebA-HQ-256 in Figure 10.

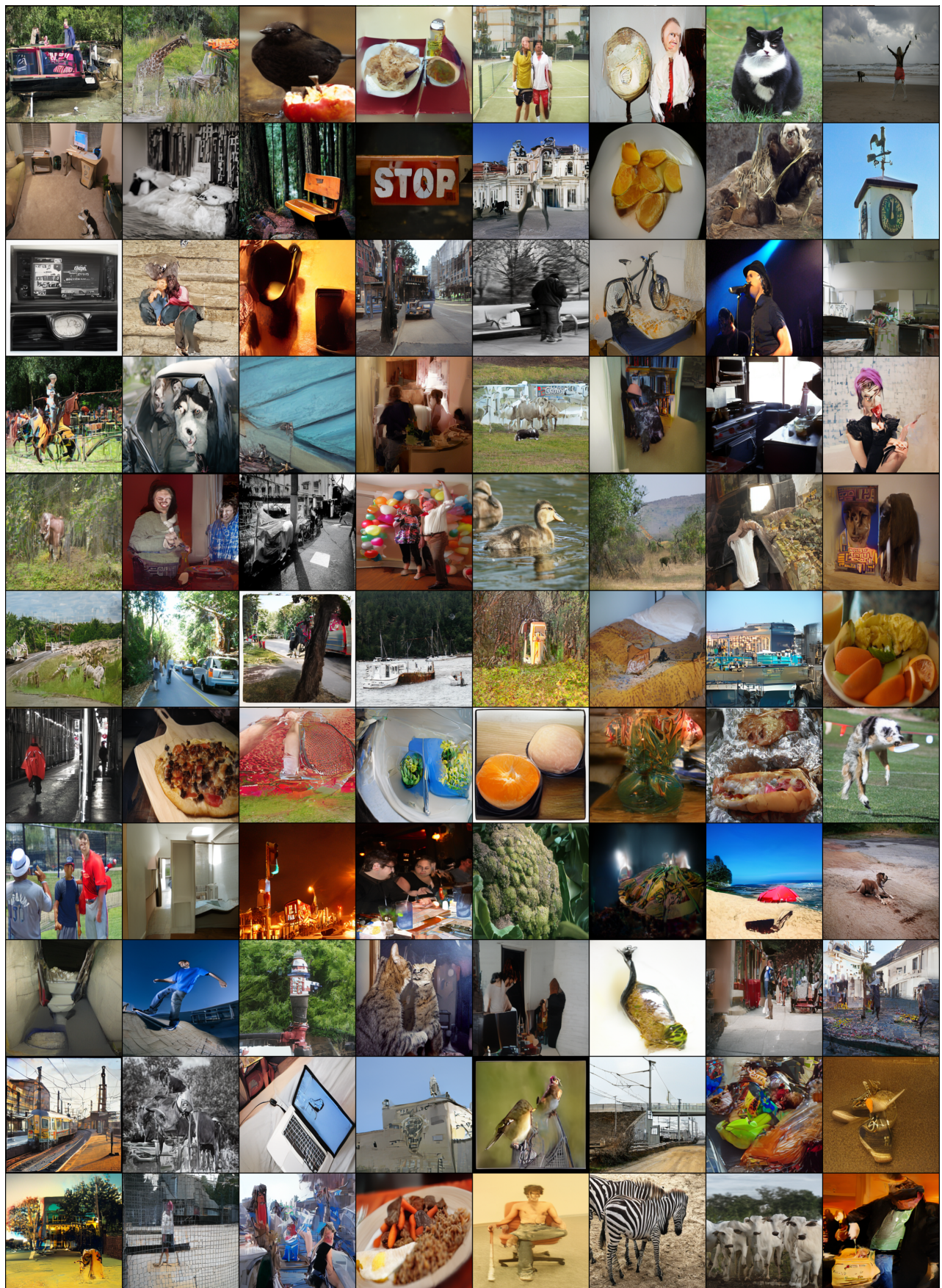


Figure 9: Uncurated generated samples of MS-COCO.



Figure 10: Uncurated generated samples of MM-CelebA-HQ-256.