CASSM: CONTENT-AWARE STATE SPACE MODELING FOR UHD LOW-LIGHT IMAGE ENHANCEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Mamba-based architectures have shown promising results in low-light image enhancement by modeling visual tokens through selective, linear-complexity scanning. However, most existing approaches rely on predefined spatial scanning strategies that determine token order mainly based on spatial locations, ignoring content similarity among visual tokens. Consequently, content-similar yet spatially distant tokens may be placed far apart in the scanning sequence, limiting effective long-range dependency modeling, particularly for ultra-high-definition (UHD) images with millions of pixels. To address this limitation, we propose a Content-Aware State Space Model (CASSM) for UHD low-light image enhancement, which constructs content-aware visual sequences with linear complexity. Specifically, CASSM leverages a learned low-frequency codebook and a lightweight feature alignment adapter to efficiently generate a content-aware assignment map. Guided by these assignments, the Content-Aware Mamba Block performs token-level content grouping to gather content-similar tokens, followed by group-level reordering using 1D t-SNE projections of the codebook entries. This hierarchical organization brings spatially distant yet visually similar tokens closer in the resulting sequence, enabling more effective state-space modeling. Extensive experiments demonstrate that CASSM achieves an average PSNR improvement of 0.75 dB over the state-of-the-art methods across four benchmarks.

1 INTRODUCTION

Ultra-high-definition (UHD) images, such as 4K and 8K resolutions, are increasingly prevalent in modern imaging systems. However, UHD images captured under low-light conditions often suffer from poor visibility, color distortion, severe noise, and detail loss, degrading both visual quality and downstream vision performance. UHD Low-Light Image Enhancement (LLIE) aims to recover well-exposed, visually pleasing images while suppressing noise and preserving or restoring structural details. CNN-based methods (Feijoo et al., 2025; Yan et al., 2025; Wang et al., 2023c) have made substantial advances in LLIE, but their ability to capture long-range dependencies is constrained by limited receptive fields, limiting restoration performance. Transformer-based methods (Xu et al., 2022; Brateanu et al., 2025; Wang et al., 2025; 2023c; Xu et al., 2023) alleviate this limitation via self-attention for long-range dependency modeling. However, quadratic self-attention complexity with respect to the token count makes them impractical for efficient UHD image enhancement.

Recently, Selective State Space Models (Mamba) (Gu & Dao, 2024) have emerged as an efficient alternative for long-range modeling with linear complexity. However, existing Mamba variants (Li et al., 2025; Lin et al., 2025; Kong et al., 2025; Wu et al., 2026) serialize visual tokens using predefined scan paths determined primarily by their spatial positions, ignoring content similarity among visual tokens. Consequently, tokens that are semantically or visually similar but spatially distant remain far apart in the sequence, making their dependencies less direct for Mamba models to capture global context, especially in UHD images containing a large number of pixels. Our key insight is that scanning sequences should instead be constructed adaptively based on token content, bringing spatially dispersed yet visually related tokens into proximity within the sequence to enable effective long-range dependency modeling while preserving linear complexity.

To realize this idea, we propose the Content-Aware State Space Model (CASSM) for UHD LLIE, which constructs content-aware visual sequences while retaining linear complexity with respect to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

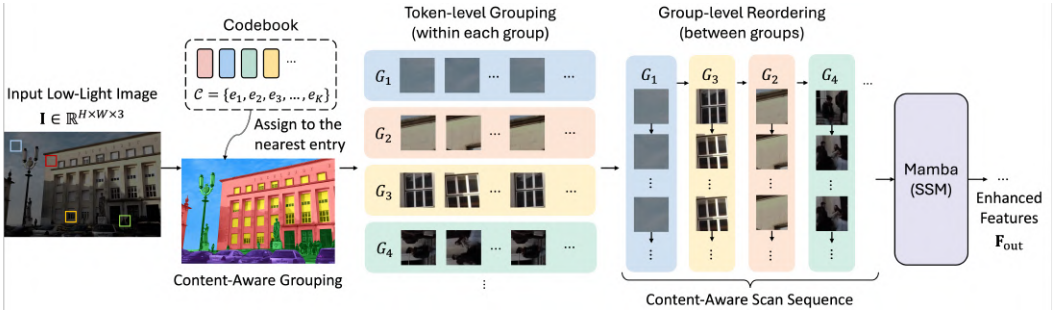


Figure 1: **Content-aware sequence organization in CASSM.** Visual tokens are grouped using their nearest codebook entries and hierarchically reorganized through token-level grouping and group-level reordering. The resulting sequence brings visually related tokens into closer proximity based on their content for effective state-space modeling.

the number of tokens. CASSM comprises three key components: (1) a low-frequency codebook that provides content anchors for efficient content grouping, (2) a Feature Alignment Adapter (FAA) that aligns restoration features with the codebook embedding space, and (3) a Content-Aware Mamba Block (CAMB) that rearranges visual tokens into content-aware sequences for state-space modeling. First, we learn a compact codebook from the low-frequency (LL) wavelet subbands, which retain principal image appearance and structures while suppressing high-frequency variations. Rather than using the codebook for feature reconstruction, we use its entries as representative content anchors to organize visual tokens by their content similarity. Second, the lightweight FAA aligns intermediate restoration features with the codebook embedding space. Each aligned token is assigned to the most similar codebook entry to generate a content-aware assignment map. Given N visual tokens and K codebook entries, this assignment requires $O(NK)$ computation. Since K is a fixed constant and $K \ll N$, the grouping process scales linearly with the sequence length. Guided by these entry assignments, CAMB hierarchically reorganizes visual tokens at two levels, as illustrated in Figure 1. Token-level content grouping gathers tokens assigned to the same codebook entry, while group-level reordering projects the codebook entries into a 1D space using t-SNE and orders the corresponding groups by their projected coordinates. By bringing spatially dispersed yet visually similar tokens closer in the sequence, CAMB can better reflect content relationships for more effective long-range state-space modeling.

Our primary contributions are summarized as follows:

- We propose CASSM, a novel Content-Aware State Space Model for UHD low-light image enhancement that adaptively organizes visual sequences according to token content while retaining linear complexity.
- We introduce a compact low-frequency codebook and a lightweight Feature Alignment Adapter for efficient content-based token assignment, providing representative content anchors for aligning restoration features with the codebook embedding space with low computational overhead.
- We develop the Content-Aware Mamba Block, which hierarchically reorganizes tokens through token-level content grouping and group-level reordering, reducing the sequence distance between spatially dispersed yet related features for effective state-space modeling.
- Extensive experiments demonstrate that CASSM achieves state-of-the-art performance on the UHD-LL, UHD-LOL4K, LOL-v1, and LOL-v2-Real datasets.

2 Related Work

2.1 Low-Light Image Enhancement

Early low-light image enhancement (LLIE) methods primarily rely on hand-crafted priors to improve visibility. For instance, Lee et al. (2013) and Xu et al. (2014) enhance contrast by adjusting intensity distributions. Wang et al. (2014) decompose images into illumination and reflectance components for illumination restoration.

With advances in deep learning, CNN-based methods (Wei et al., 2018; Yang et al., 2021; Wang et al., 2023a; Zhang et al., 2024; Feijoo et al., 2025; Yan et al., 2025; Wang et al., 2023c) have achieved substantial progress in LLIE. Retinex-Net (Wei et al., 2018) and SGM-Net (Yang et al., 2021) incorporate Retinex decomposition into neural networks, while FourLLIE (Wang et al., 2023a) and DMFourLLIE (Zhang et al., 2024) leverage frequency-domain representations to improve illumination and detail recovery. Despite these advances, CNN-based architectures remain constrained by their limited receptive fields. Transformer-based methods (Xu et al., 2022; Cai et al., 2023; Wang et al., 2025; 2023c; Xu et al., 2023) address this limitation by leveraging self-attention to model long-range dependencies. For instance, Xu et al. (2022) employ an SNR-aware Transformer to perform spatially varying enhancement, while Retinexformer (Cai et al., 2023) introduces illumination-guided attention to facilitate interactions between regions with different exposure levels. However, the quadratic complexity of self-attention with respect to the number of tokens makes these methods intractable for ultra-high-definition (UHD) images.

UHD image restoration (Wang et al., 2023b; Liu et al., 2025; Wang et al., 2026; Yu et al., 2023; Li et al., 2023; Wu et al., 2025; Wang et al., 2024) has emerged as an active research topic in recent years. For instance, Wang et al. (2023b) introduce axis-based attention to reduce computational complexity. Liu et al. (2025) perform restoration in a compact VAE latent space. Wang et al. (2026) model the interactions between gradient and normal priors in a low-resolution feature space. These studies highlight the importance of efficient architectural design for UHD LLIE and motivate us to explore a more effective approach to long-range dependency modeling.

2.2 State Space Models for Image Restoration

Selective State Space Models (Mamba) (Gu & Dao, 2024) have emerged as an efficient alternative for long-range modeling, owing to their linear computational complexity. VisionMamba (Zhu et al., 2024b) first adapts Mamba to visual representation learning by flattening 2D image patches into 1D sequences and applying bidirectional selective scanning to capture global context. Recently, several studies (Guo et al., 2024; Liu et al., 2024; Li et al., 2025; Lin et al., 2025; Kong et al., 2025; Wu et al., 2026; Zou et al., 2024; Guo et al., 2025) have applied Mamba to image restoration. For instance, EAMamba (Lin et al., 2025) develops a multi-head selective scan module with an all-around scanning mechanism to alleviate local pixel forgetting. Wave-Mamba (Zou et al., 2024) performs state-space modeling in the wavelet domain. MambaIRv2 (Guo et al., 2025) employs semantic prompts and learned routing to incorporate semantic information into state-space modeling. However, most Mamba-based methods primarily rely on fixed spatial scanning strategies within the spatial domain, ignoring input-specific content correlations among visual tokens. Consequently, visually similar yet spatially distant tokens may remain far apart in the resulting sequence, limiting Mamba's ability to capture global context, particularly in UHD images containing millions of pixels.

The work most closely related to ours is C^2SSM (Wu et al., 2026), which clusters pixels before scanning. However, it compresses millions of tokens into four centroids and applies Mamba only to them, limiting fine-grained token interactions and underutilizing Mamba's modeling capacity. Moreover, the centroid ordering does not consider content similarity, separating related clusters in the scanning sequence.

To address these limitations, our CASSM constructs content-aware scanning sequences by explicitly reorganizing tokens according to their similarity. This brings spatially dispersed yet visually related tokens closer in the sequence, enabling effective long-range dependency modeling while maintaining linear computational complexity.

3 Proposed Method

We propose the Content-Aware State Space Model (CASSM), a novel Mamba-based framework tailored for UHD low-light image enhancement. Unlike conventional methods that construct visual sequences using predefined spatial scans, CASSM reorganizes visual tokens according to content similarity while retaining linear complexity. In what follows, we present the overall CASSM framework and then elaborate on its core components.

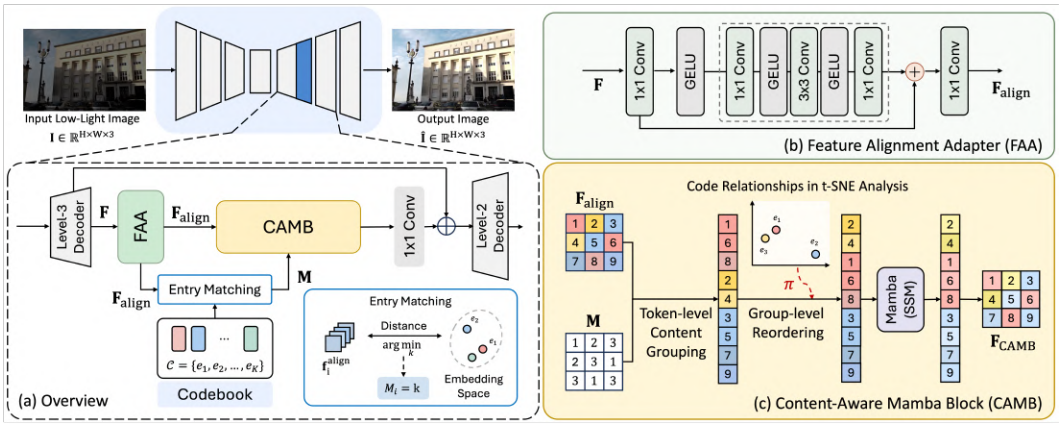


Figure 2: **Overview of our CASSM framework.** (a) Given a low-light image, the encoder–decoder backbone extracts hierarchical features. At the Level-3 decoder, the Feature Alignment Adapter (FAA) and Content-Aware Mamba Block (CAMB) process the decoder feature F with content-aware state-space modeling before propagating it to Level 2. (b) The FAA maps intermediate backbone features F into the codebook embedding space for reliable entry assignment. (c) The CAMB constructs content-aware sequences via token-level content grouping and group-level reordering, bringing visually related tokens closer in the sequence before selective state-space modeling.

3.1 Overall Framework

As illustrated in Figure 2(a), we adopt the three-level encoder–decoder architecture of C^{2}SSM (Wu et al., 2026) as our backbone. Given a low-light UHD image $\mathbf{I}$, the encoder extracts hierarchical representations, which are subsequently reconstructed by the decoder. We attach CASSM to the Level-3 decoder for content-aware feature modeling. CASSM comprises three components: a **low-frequency codebook** for content-based token grouping, a **Feature Alignment Adapter (FAA)** that aligns backbone features with the codebook space, and a **Content-Aware Mamba Block (CAMB)** that compiles and models content-aware sequences. The output of CAMB is projected through a 1×1 convolution and residually fused with the original decoder feature. The fused feature is then passed through the remaining decoder levels to predict an image residual, which is added to $\mathbf{I}$ to produce the enhanced image $\widehat{\mathbf{I}}$. Given the corresponding normal-light ground-truth image $\mathbf{I}_{\mathrm{gt}}$, the entire network is optimized using a combined spatial and frequency reconstruction objective:

$$\mathcal{L}_{\mathrm{total}} = \left\| \widehat{\mathbf{I}} - \mathbf{I}_{\mathrm{gt}} \right\|_1 + \lambda \left\| \mathcal{F}(\widehat{\mathbf{I}}) - \mathcal{F}(\mathbf{I}_{\mathrm{gt}}) \right\|_1, \tag{1}$$

where $\mathcal{F}(\cdot)$ denotes the two-dimensional Fourier transform and λ controls the contribution of FFT loss (Wu et al., 2026).

3.2 Low-Frequency Codebook Learning

CASSM aims to reorganize visual tokens by content similarity while preserving linear complexity. Direct pairwise matching across N tokens incurs $O(N^2)$ complexity and is impractical for UHD images. We therefore introduce a compact low-frequency codebook with K entries, allowing us to assign N tokens to K content anchors in $O(NK)$ time. Since K is fixed and $K \ll N$, the assignment process scales linearly with the number of tokens. Notably, unlike conventional codebook-based methods that use discrete codes directly for feature reconstruction, we exploit the resulting assignments to guide content-aware sequence modeling.

During codebook training, we randomly sample a 256×256 patch from each high-quality image at a time and decompose it into four subbands using the Discrete Wavelet Transform (DWT) (Mallat, 1989). We retain its low-frequency (LL) subband, denoted as $\mathbf{X}_{LL}$, for codebook learning. This LL subband preserves the dominant structural geometry and global illumination characteristics while suppressing high-frequency variations such as fine textures and noise, providing stable references for content grouping. To facilitate stable training, we normalize $\mathbf{X}_{LL}$ using the channel-wise statistics

over the training set as:

$$\widetilde{\mathbf{X}}_{LL} = \mathrm{clip}\left(\frac{\mathbf{X}_{LL} - \boldsymbol{\mu}}{3\boldsymbol{\sigma} + \epsilon}, -1, 1\right), \tag{2}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ denote the channel-wise mean and standard deviation computed over the training set, respectively, and ϵ is a small constant for numerical stability. The normalized LL patch is encoded into a latent representation $\mathbf{Z} = E(\widetilde{\mathbf{X}}_{LL}) \in \mathbb{R}^{h\times w\times D}$, where E denotes the VQ-VAE encoder (Van Den Oord et al., 2017), h and w are the latent spatial dimensions, and D is the embedding dimension.

We define a learnable codebook $\mathcal{C} = \{\mathbf{e}_k\}_{k=1}^{K}$, where $\mathbf{e}_k \in \mathbb{R}^D$ denotes the k-th codebook entry, and K is the codebook size. Following the vector quantization scheme of VQ-VAE, each latent vector $\mathbf{z}_i \in \mathbb{R}^D$ in $\mathbf{Z}$ at spatial index i is mapped to its nearest codebook entry in $\mathcal{C}$. To prevent the magnitude bias from dominating the matching criterion, both the latent vectors and codebook entries are ℓ_2-normalized when computing their distance. The quantized vector is obtained as

$$\mathcal{Q}(\mathbf{z}_i) = \arg\min_{\mathbf{e}_k\in\mathcal{C}} \left\| \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2} - \frac{\mathbf{e}_k}{\|\mathbf{e}_k\|_2} \right\|_2^2 = \arg\max_{\mathbf{e}_k\in\mathcal{C}} \cos(\mathbf{z}_i, \mathbf{e}_k). \tag{3}$$

where $\mathcal{Q}(\cdot)$ denotes the vector quantization operator that assigns $\mathbf{z}_i$ to the codebook entry with the smallest distance in the normalized embedding space, which is equivalent to selecting the codebook entry with the highest cosine similarity. Applying $\mathcal{Q}(\cdot)$ individually to all spatial locations of $\mathbf{Z}$ produces the quantized latent representation $\mathbf{Z_q}$. The VQ-VAE decoder G subsequently reconstructs the normalized LL subband from $\mathbf{Z_q}$:

$$\widehat{\mathbf{X}}_{LL} = G(\mathbf{Z_q}) = G\left(\mathcal{Q}(\mathbf{Z})\right). \tag{4}$$

Following VQ-VAE, the codebook is optimized using a reconstruction loss and a commitment loss:

$$\mathcal{L}_{LL} = \left\| \widehat{\mathbf{X}}_{LL} - \widetilde{\mathbf{X}}_{LL} \right\|_2^2 + \beta \left\| \mathrm{sg}[\mathbf{Z}_q] - \mathbf{Z} \right\|_2^2, \tag{5}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator and β controls the commitment weighting. After training, we retain only the learned codebook $\mathcal{C}$ as a frozen set of content anchors for subsequent content grouping. The VQ-VAE encoder and decoder are discarded as required only for codebook learning. VQ-VAE architecture, codebook update, and training details are provided in Appendix A.2.

3.3 Content-Aware Modeling

With the learned codebook providing content guidance, the proposed CASSM performs content-aware sequence construction and state-space modeling via two cascading modules: the feature alignment adapter (FAA) and the Content-Aware Mamba Block (CAMB).

3.3.1 Feature Alignment Adapter

Given a low-light image $\mathbf{I}$, the image encoder $\mathcal{E}$ and the Level-3 decoder $\mathcal{D}_3$ produce an intermediate restoration feature:

$$\mathbf{F} = \mathcal{D}_3\big(\mathcal{E}(\mathbf{I})\big) \in \mathbb{R}^{H_f\times W_f\times C_f}, \tag{6}$$

where H_f, W_f, and C_f denote its height, width, and channel dimension, respectively. Because $\mathbf{F}$ originates from the restoration backbone while the codebook $\mathcal{C}$ is learned in a latent embedding space derived from LL subbands, these two representations are not inherently aligned, and directly matching them may lead to unreliable token assignments. We therefore introduce a Feature Alignment Adapter (FAA) that maps $\mathbf{F}$ into the codebook embedding space to bridge the representation gap:

$$\mathbf{F}_{\text{align}} = \mathrm{FAA}(\mathbf{F}) \in \mathbb{R}^{H_f\times W_f\times D}, \tag{7}$$

where D is the embedding dimension of the codebook entries. As illustrated in Figure 2(b), FAA consists of stacked 1×1 and 3×3 convolutions with GELU activations and a residual connection.

For each aligned token $\mathbf{f}_i^{\text{align}} \in \mathbb{R}^D$ at spatial index $i \in \{1,\ldots,N\}$, where $N = H_f W_f$, we assign it to the codebook entry with the highest cosine similarity:

$$M_i = \arg\max_{k\in\{1,\ldots,K\}} \cos\left(\mathbf{f}_i^{\text{align}}, \mathbf{e}_k\right), \tag{8}$$

where $M_i \in \{1, \ldots, K\}$ denotes the index of the codebook entry assigned to the i-th token. Applying this matching operation to all N tokens yields the content-aware assignment map $\mathbf{M} = \{M_i\}_{i=1}^{N}$, which guides the subsequent sequence reorganization in CAMB.

3.3.2 Content-Aware Mamba Block

Given the aligned feature $\mathbf{F}_{\text{align}}$ and the assignment map $\mathbf{M}$, the Content-Aware Mamba Block (CAMB) reorganizes visual tokens into content-aware 1D sequences for state-space modeling, as illustrated in Figure 2(c). CAMB comprises three sequential stages: token-level content grouping, group-level reordering, and content-aware sequence modeling, as detailed below.

Token-Level Content Grouping. We first cluster the aligned tokens according to their assigned codebook indices. Specifically, tokens mapped to the same codebook entry are gathered into a contiguous content group, regardless of their original spatial locations in the 2D feature map. Regarding the k-th codebook entry, its ordered index sequence $\mathcal{I}_k$ is defined as:

$$\mathcal{I}_k = (i_{k,1}, i_{k,2}, \ldots, i_{k,n_k}),\ M_{i_{k,j}} = k, \tag{9}$$

where n_k denotes the number of tokens assigned to the k-th codebook entry, and the indices in $\mathcal{I}_k$ follow their original raster-scan order ($i_{k,1} < i_{k,2} < \cdots < i_{k,n_k}$). The corresponding token content group $\mathcal{G}_k$ is constructed as:

$$\mathcal{G}_k = \left(\mathbf{f}^{\text{align}}_{i_{k,1}}, \mathbf{f}^{\text{align}}_{i_{k,2}}, \ldots, \mathbf{f}^{\text{align}}_{i_{k,n_k}}\right). \tag{10}$$

This grouping yields a collection of K token groups, $\{\mathcal{G}_k\}_{k=1}^{K}$, arranging potentially spatially dispersed tokens with the same codebook assignment into contiguous sequence segments while preserving their intra-group raster-scan order. Since some groups may be empty if no token is assigned to the corresponding codebook entry, we retain all K groups in the formulation for notational simplicity, with empty groups contributing no tokens to subsequent sequence construction.

Group-Level Reordering. While token-level content grouping brings tokens with the same codebook assignment together, the ordering among the resulting groups $\{\mathcal{G}_k\}$ still follows their arbitrary codebook indices and do not reflect their content relationships. To establish coherent transitions along the sequence, we reorder these groups based on local similarity relationships among the codebook entries. We employ t-SNE (van der Maaten & Hinton, 2008) to embed the codebook entries into a low-dimensional space such that entries with similar representations tend to remain close to each other. Specifically, after codebook training, we project all codebook entries $\{\mathbf{e}_k\}_{k=1}^{K}$ into a 1D manifold via t-SNE:

$$(r_1, \ldots, r_K) = \text{t-SNE}_{\text{1D}}(\mathbf{e}_1, \ldots, \mathbf{e}_K), \tag{11}$$

where $r_k \in \mathbb{R}$ denotes the 1D embedding coordinate of the k-th codebook entry $\mathbf{e}_k$. The group ordering $\boldsymbol{\pi}$ is determined by sorting these coordinates:

$$\boldsymbol{\pi} = \text{argsort}(r_1, \ldots, r_K), \tag{12}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ specifies the resulting ordering of the codebook indices. Since the codebook remains fixed during restoration training and inference, this group ordering is precomputed offline once and introduces no image-dependent projection overhead. Following the group permutation $\boldsymbol{\pi}$, we retain the non-empty groups in the resulting order and concatenate them to form the final content-aware sequence:

$$\mathbf{S} = \text{Concat}(\mathcal{G}_{\pi_1}, \mathcal{G}_{\pi_2}, \ldots, \mathcal{G}_{\pi_K}), \tag{13}$$

where tokens preserve their original raster-scan order within each group.

Content-Aware Sequence Modeling. After token-level content grouping and group-level reordering, the resulting content-aware sequence $\mathbf{S}$ is processed as follows:

$$\begin{aligned} \mathbf{S}' &= \mathbf{S} + \text{S6}\big(\text{Norm}(\mathbf{S})\big), \\ \mathbf{S}_{\text{out}} &= \mathbf{S}' + \text{FFN}\big(\text{Norm}(\mathbf{S}')\big), \\ \mathbf{F}_{\text{out}} &= \mathbf{F} + \text{Conv}_{1\times1}\left(\text{Reshape}\left(\text{Restore}(\mathbf{S}_{\text{out}})\right)\right). \end{aligned} \tag{14}$$

Here, $\text{S6}(\cdot)$ denotes the selective state-space operator (Gu & Dao, 2024), while $\text{Norm}(\cdot)$ and $\text{FFN}(\cdot)$ represent normalization and the feed-forward network, respectively. $\text{Restore}(\cdot)$ reorders the modeled tokens according to their original sequence indices retained during content grouping, while

Table 1: Comparison of quantitative results on the UHD-LOL4K dataset.

Methods	Venue	PSNR	SSIM	Params (M)
Restormer	CVPR'22	36.91	0.988	26.11
DiffLL	SIGGRAPH'23	36.95	0.989	17.29
NSEN	MM'23	29.49	0.980	2.67
UHDFour	ICLR'23	36.12	<u>0.990</u>	17.54
LLFormer	AAAI'23	37.33	0.988	13.15
Wave-Mamba	MM'24	37.43	<u>0.990</u>	1.25
UHDformer	AAAI'24	36.28	0.989	0.34
D2Net	WACV'25	37.73	**0.992**	5.22
C^{2}SSM	CVPR'26	<u>39.61</u>	**0.992**	2.71
CASSM (ours)	–	**40.93**	**0.992**	3.95

Table 2: Comparison of quantitative results on the UHD-LL dataset.

Methods	Venue	PSNR	SSIM	Params (M)
Restormer	CVPR'22	23.46	0.906	26.11
DiffLL	SIGGRAPH'23	21.36	0.872	17.29
UHDFour	ICLR'23	26.22	0.900	17.54
LLFormer	AAAI'23	22.79	0.853	13.15
UHDformer	AAAI'24	27.11	0.927	0.34
Wave-Mamba	MM'24	<u>27.35</u>	0.913	1.25
UHDDIP	TCSVT'25	26.74	0.928	0.81
UHD-processer	CVPR'25	27.22	<u>0.929</u>	1.60
C^{2}SSM	CVPR'26	27.14	**0.931**	2.71
CASSM (ours)	–	**27.43**	**0.931**	3.95

Table 3: Comparison of quantitative results on LOL-v2-Real and LOL-v1 datasets.

Methods	Venue	LOL-v2-Real		LOL-v1		Params (M)	FLOPs (G)
		PSNR	SSIM	PSNR	SSIM		
FourLLIE	MM'23	23.45	0.845	20.99	0.807	0.12	4.07
Retinexformer	ICCV'23	24.55	0.843	22.71	0.817	1.61	15.57
UHDFour	ICLR'23	27.27	0.857	22.89	0.819	17.54	4.78
LLFormer	AAAI'23	27.75	0.861	<u>23.65</u>	0.816	24.52	11.26
DMFourLLIE	MM"24	26.40	0.876	22.98	0.827	0.41	1.70
Wave-Mamba	MM'24	27.87	0.893	22.76	0.842	1.25	7.22
RetinexMamba	ICONIP'24	27.31	0.866	23.15	0.821	3.59	34.76
CWNet	ICCV'25	27.39	0.900	23.60	<u>0.849</u>	1.23	11.3
C^{2}SSM	CVPR'26	<u>31.30</u>	<u>0.911</u>	23.40	**0.853**	2.71	0.407
CASSM (ours)	–	**32.64**	**0.919**	**23.89**	**0.853**	3.95	0.449

$\mathrm{Reshape}(\cdot)$ converts the restored sequence into a spatial feature map. Finally, the resulting feature map is projected through a 1×1 convolution and residually fused with the original backbone feature $\mathbf{F}$ to produce the CAMB output $\mathbf{F}_{\mathrm{out}}$. By organizing visually related tokens closer in the sequence, CAMB enables more effective long-range state-space modeling.

4 Experiments

4.1 Experimental Settings

Implementation Details. The CASSM training procedure comprises two stages. First, we pretrain the low-frequency codebook using high-quality images from five UHD restoration training sets: UHD-LL (Li et al., 2023), UHD-Haze and UHD-Blur (Wang et al., 2024), and UHD-Rain and UHD-Snow (Wang et al., 2026). The codebook comprises $K = 1024$ entries with an embedding dimension of $D = 128$ and is trained for 400K iterations using randomly cropped 256×256 patches. Next, we adopt the encoder–decoder architecture of C^{2}SSM as the backbone. We keep the codebook fixed while training CASSM for 150K iterations using 768×768 patches. Both stages use AdamW with a batch size of 16 and an initial learning rate of 5×10^{-4}. During training and inference, the fixed codebook provides content-grouping guidance without parameter updates. Our experiments are conducted on four NVIDIA GeForce RTX 4090 GPUs.

Datasets and Evaluation Protocol. We conduct experiments on two UHD benchmarks, UHD-LL (Li et al., 2023) and UHD-LOL4K (Wang et al., 2023b), and two standard-resolution benchmarks, LOL-v1 (Wei et al., 2018) and LOL-v2-Real (Yang et al., 2021). UHD-LL consists of 2,000 training pairs and 150 testing pairs, while UHD-LOL4K provides 5,999 training pairs and 2,100 testing pairs. LOL-v1 contains 485 training pairs and 15 testing pairs, and the LOL-v2-Real test set contains 100 pairs. For UHD-LL, UHD-LOL4K, and LOL-v1, we train and evaluate the models using their respective training and testing splits. LOL-v2-Real is used exclusively for cross-dataset evaluation, where the model is trained on LOL-v1 without further training or adaptation.

Compared Methods. We compare CASSM with representative LLIE methods, including Restormer (Zamir et al., 2022), FourLLIE (Wang et al., 2023a), DiffLL (Jiang et al., 2023), DM-FourLLIE (Zhang et al., 2024), Retinexformer (Cai et al., 2023), RetinexMamba (Bai et al., 2024), and CWNet (Zhang et al., 2025). We further compare with methods designed for UHD image enhancement, including NSEN (Yu et al., 2023), UHDFour (Li et al., 2023), LLFormer (Wang et al.,

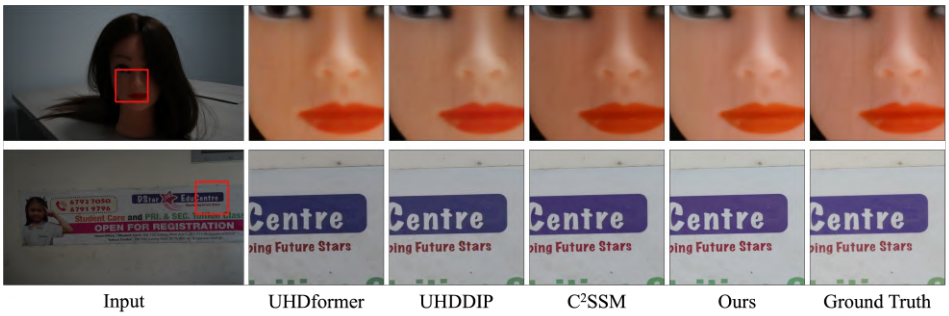


Figure 3: Qualitative comparisons on the UHD-LL dataset.

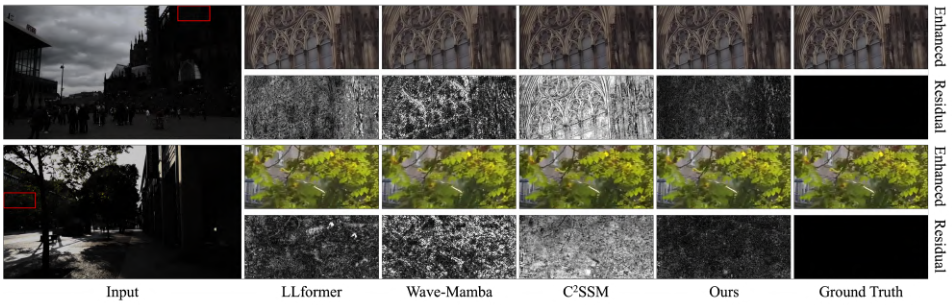


Figure 4: Qualitative comparisons on the UHD-LOL4K dataset. *Enhanced* and *Residual* denote the enhanced images and their residual maps with respect to the ground truth, respectively.

2023b), UHDformer (Wang et al., 2024), Wave-Mamba (Zou et al., 2024), UHD-processer (Liu et al., 2025), D2Net (Wu et al., 2025), UHDDIP (Wang et al., 2026), and C^{2}SSM (Wu et al., 2026).

4.2 Experimental Results

Quantitative Results on UHD Benchmarks. Tables 1 and 2 compare CASSM with representative methods on the UHD-LOL4K and UHD-LL datasets, respectively. On UHD-LOL4K, CASSM achieves the best PSNR of 40.93 dB, surpassing the strong C^{2}SSM baseline by 1.32 dB. This improvement demonstrates the effectiveness of content-aware sequence organization for long-range dependency modeling in high-resolution images. On UHD-LL, CASSM also obtains the highest PSNR of 27.43 dB, outperforming UHD-processer and C^{2}SSM by 0.21 dB and 0.29 dB, respectively. Overall, CASSM achieves state-of-the-art performance on both UHD benchmarks.

Quantitative Results on Standard-Resolution Benchmarks. Table 3 presents the quantitative comparisons on the LOL-v2-Real and LOL-v1 benchmarks. On the cross-dataset LOL-v2-Real evaluation, CASSM achieves the highest PSNR of 32.64 dB, outperforming the previous best method, C^{2}SSM, by 1.34 dB. This substantial improvement demonstrates the strong generalization of content-aware sequence modeling to unseen low-light data. On LOL-v1, CASSM also achieves the highest PSNR of 23.89 dB, exceeding LLFormer by 0.24 dB and C^{2}SSM by 0.49 dB. These results demonstrate that CASSM achieves strong in-domain performance and cross-dataset generalization with modest computational overhead.

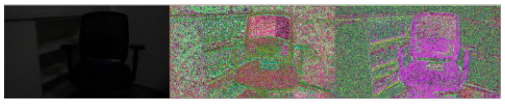
Qualitative Results. Figures 3 and 4 present qualitative comparisons on UHD-LL and UHD-LOL4K, respectively. The residual maps visualize differences from the ground truth, with lower intensities indicating smaller errors. CASSM better restores color and structural details on UHD-LL while producing cleaner results with lower residual errors on UHD-LOL4K. Overall, these results demonstrate more faithful detail preservation and brightness recovery.

4.3 Ablation Study

To validate the effectiveness of each proposed component, we conduct a series of ablation experiments on the UHD-LOL4K dataset. All models are trained for 75K training iterations while keeping the remaining training settings unchanged. Specifically, we evaluate the effectiveness of individual components in CASSM, analyze the effectiveness of content-aware sequence construction, compare with different Mamba designs, and analyze content-aware grouping.

Table 4: Effectiveness of each component.

FAA	CAMB	PSNR	SSIM
×	×	38.51	0.9897
✓	×	38.59	0.9896
×	✓	38.83	0.9902
✓	✓	38.99	0.9903



Input w/o FAA w/ FAA

Figure 5: Analysis of content-aware grouping.

Table 5: Effectiveness of content-aware sequence construction.

Token-Level Grouping	Group-Level Reordering	PSNR	SSIM
×	×	38.68	0.9896
✓	×	38.76	0.9899
✓	✓	38.99	0.9903

Table 6: Comparison with different Mamba designs.

Method	EVSSM	EAMamba	CASSM (ours)
PSNR	38.46	38.85	38.99
SSIM	0.9891	0.9897	0.9903

Effectiveness of Individual Components. As shown in Table 4, introducing the Adapter or CAMB individually improves the baseline PSNR from 38.51 dB to 38.59 dB and 38.83 dB, respectively. When combining the Adapter with CAMB, the performance is further improved to 38.99 dB, outperforming CAMB alone by 0.16 dB. These results demonstrate that feature alignment further facilitates the subsequent content-aware modeling in CAMB.

Effectiveness of Content-Aware Sequence Construction. Table 5 evaluates the contributions of token-level content grouping and group-level reordering. The baseline adopts conventional spatial sequence construction. Token-level grouping improves the PSNR to 38.76 dB, where the groups are concatenated according to their original codebook indices. Further applying group-level reordering achieves 38.99 dB, demonstrating the benefit of content organization at both pixel and group levels.

Comparison with Different Mamba Designs. For a fair comparison with different Mamba-based designs, we replace the entire CASSM module with representative Mamba blocks, including EVSSM and EAMamba, at the same insertion point while keeping the backbone and all other components unchanged. Table 6 shows that EVSSM and EAMamba achieve 38.46 dB and 38.85 dB, respectively, while CASSM achieves 38.99 dB. These results demonstrate the effectiveness of the proposed content-aware sequence modeling over alternative Mamba blocks under the same setting.

Analysis of Content-Aware Grouping. Figure 5 visualizes codebook assignments with and without FAA, where tokens assigned to the same entry share the same color. Without FAA, the assignments are noisy and fragmented. In contrast, FAA produces coherent regions that better correspond to image content, demonstrating that CASSM can effectively group content-related tokens even under extremely low illumination.

5 Limitations

CASSM relies on a fixed low-frequency codebook learned from the training data. Therefore, its entries may not fully capture content distributions that differ substantially from those in the training data, potentially leading to suboptimal assignments for unseen features. Future work could explore dynamically adapting or expanding the codebook to better address such distribution shifts while preserving computational efficiency.

6 Conclusion

We propose CASSM, a novel Content-Aware State Space Model for UHD low-light image enhancement. Unlike existing Mamba-based methods that construct visual sequences using predefined spatial scans, CASSM reorders visual tokens according to input-specific content correlations while preserving linear complexity. To efficiently identify related features without costly pairwise comparisons, we learn a compact codebook from low-frequency wavelet subbands and introduce a lightweight Feature Alignment Adapter to align intermediate backbone features with the codebook space for reliable content assignment. Guided by these assignments, the proposed Content-Aware Mamba Block hierarchically reorganizes visual tokens through token-level content grouping and group-level reordering, bringing spatially dispersed yet related features closer in the resulting sequence for effective state-space modeling. Experimental results demonstrate that CASSM achieves state-of-the-art performance across four benchmarks, outperforming previous methods by an average of 0.75 dB in PSNR.

REFERENCES

Jiesong Bai, Yuhao Yin, Qiyuan He, Yuanxian Li, and Xiaofeng Zhang. RetinexMamba: Retinex-based mamba for low-light image enhancement. In *International Conference on Neural Information Processing*, pp. 427–442, 2024.

Alexandru Brateanu, Raul Balmez, Adrian Avram, Ciprian Orhei, and Cosmin Ancuti. LYT-NET: Lightweight YUV transformer-based network for low-light image enhancement. *IEEE Signal Processing Letters*, 32:2065–2069, 2025.

Yuanhao Cai, Hao Bian, Jing Lin, Haoqian Wang, Radu Timofte, and Yulun Zhang. Retinexformer: One-stage retinex-based transformer for low-light image enhancement. In *IEEE/CVF International Conference on Computer Vision*, pp. 12504–12513, 2023.

Daniel Feijoo, Juan C Benito, Alvaro Garcia, and Marcos V Conde. Darkir: Robust low-light image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10879–10889, 2025.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2024.

Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. MambaIR: A simple baseline for image restoration with state-space model. In *European Conference on Computer Vision*, pp. 222–241, 2024.

Hang Guo, Yong Guo, Yaohua Zha, Yulun Zhang, Wenbo Li, Tao Dai, Shu-Tao Xia, and Yawei Li. MambaIRv2: Attentive state space restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28124–28133, 2025.

Hai Jiang, Ao Luo, Songchen Han, Haoqiang Fan, and Shuaicheng Liu. Low-light image enhancement with wavelet-based diffusion models. *ACM Transactions on Graphics*, 42:1–14, 2023.

Lingshun Kong, Jiangxin Dong, Jinhui Tang, Ming-Hsuan Yang, and Jinshan Pan. Efficient visual state space model for image deblurring. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12710–12719, 2025.

Chulwoo Lee, Chul Lee, and Chang-Su Kim. Contrast enhancement based on layered difference representation of 2d histograms. *IEEE Transactions on Image Processing*, 22:5372–5384, 2013.

Boyun Li, Haiyu Zhao, Wenxin Wang, Peng Hu, Yuanbiao Gou, and Xi Peng. MaIR: A locality- and continuity-preserving mamba for image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7491–7501, 2025.

Chongyi Li, Chun-Le Guo, Man Zhou, Zhexin Liang, Shangchen Zhou, Ruicheng Feng, and Chen Change Loy. Embedding fourier for ultra-high-definition low-light image enhancement. In *International Conference on Learning Representations*, 2023.

Yu-Cheng Lin, Yu-Syuan Xu, Hao-Wei Chen, Hsien-Kai Kuo, and Chun-Yi Lee. EAMamba: Efficient all-around vision state space model for image restoration. In *IEEE/CVF International Conference on Computer Vision*, pp. 11708–11719, 2025.

Kechun Liu, Yitong Jiang, Inchang Choi, and Jinwei Gu. Learning image-adaptive codebooks for class-agnostic image restoration. In *IEEE/CVF International Conference on Computer Vision*, pp. 5373–5383, 2023.

Yidi Liu, Dong Li, Xueyang Fu, Xin Lu, Jie Huang, and Zheng-Jun Zha. UHD-processer: Unified UHD image restoration with progressive frequency learning and degradation-aware prompts. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23121–23130, 2025.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. VMamba: Visual state space model. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

S.G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

Fumin Shen, Chunhua Shen, Qinfeng Shi, Anton van den Hengel, and Zhenmin Tang. Inductive hashing on manifolds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

Chenxi Wang, Hongjun Wu, and Zhi Jin. FourLLIE: Boosting low-light image enhancement by fourier frequency information. In *ACM International Conference on Multimedia*, pp. 7459–7469, 2023a.

Cong Wang, Jinshan Pan, Wei Wang, Gang Fu, Siyuan Liang, Mengzhu Wang, Xiao-Ming Wu, and Jun Liu. Correlation matching transformation transformers for UHD image restoration. In *AAAI Conference on Artificial Intelligence*, volume 38, pp. 5336–5344, 2024.

Hongzhi Wang, Xiubo Liang, Jinxing Han, and Weidong Geng. Spike-retinexformer: Rethinking low-light image enhancement with spiking neural networks. *Advances in Neural Information Processing Systems*, 38:11425–11444, 2025.

Liqian Wang, Liang Xiao, Hongyi Liu, and Zhihui Wei. Variational bayesian method for retinex. *IEEE Transactions on Image Processing*, 23:3381–3396, 2014.

Liyan Wang, Cong Wang, Jinshan Pan, Xiaofeng Liu, Weixiang Zhou, Xiaoran Sun, Wei Wang, and Zhixun Su. Ultra-high-definition image restoration: New benchmarks and a dual interaction prior-driven solution. *IEEE Transactions on Circuits and Systems for Video Technology*, 36:2052–2068, 2026.

Tao Wang, Kaihao Zhang, Tianrun Shen, Wenhan Luo, Bjorn Stenger, and Tong Lu. Ultra-high-definition low-light image enhancement: A benchmark and transformer-based method. In *AAAI Conference on Artificial Intelligence*, volume 37, pp. 2654–2662, 2023b.

Yinglong Wang, Zhen Liu, Jianzhuang Liu, Songcen Xu, and Shuaicheng Liu. Low-light image enhancement with illumination-aware gamma correction and complete image modelling network. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13082–13091, 2023c.

Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *British Machine Vision Conference*, 2018.

Chen Wu, Ling Wang, Long Peng, Dianjie Lu, and Zhuoran Zheng. Dropout the high-rate down-sampling: A novel design paradigm for UHD image restoration. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2390–2399, 2025.

Chen Wu, Ling Wang, Zhuoran Zheng, Yuning Cui, Zhixiong Yang, Xiangyu Chen, Yue Zhang, Weidong Jiang, and Jingyuan Xia. Scan clusters, not pixels: A cluster-centric paradigm for efficient ultra-high-definition image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15528–15537, 2026.

Hongteng Xu, Guangtao Zhai, Xiaolin Wu, and Xiaokang Yang. Generalized equalization model for image enhancement. *IEEE Transactions on Multimedia*, 16:68–82, 2014.

Xiaogang Xu, Ruixing Wang, Chi-Wing Fu, and Jiaya Jia. SNR-aware low-light image enhancement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17693–17703, 2022.

Xiaogang Xu, Ruixing Wang, and Jiangbo Lu. Low-light image enhancement via structure modeling and guidance. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9893–9903, 2023.

Qingsen Yan, Yixu Feng, Cheng Zhang, Guansong Pang, Kangbiao Shi, Peng Wu, Wei Dong, Jinqiu Sun, and Yanning Zhang. Hvi: A new color space for low-light image enhancement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5678–5687, 2025.

Wenhan Yang, Wenjing Wang, Haofeng Huang, Shiqi Wang, and Jiaying Liu. Sparse gradient regularized deep retinex network for robust low-light image enhancement. *IEEE Transactions on Image Processing*, 30:2072–2086, 2021.

Wei Yu, Qi Zhu, Naishan Zheng, Jie Huang, Man Zhou, and Feng Zhao. Learning non-uniform-sampling for ultra-high-definition image enhancement. In *ACM International Conference on Multimedia*, pp. 1412–1421, 2023.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5728–5739, 2022.

Tongshun Zhang, Pingping Liu, Ming Zhao, and Haotian Lv. DMFourLLIE: Dual-stage and multi-branch fourier network for low-light image enhancement. In *ACM International Conference on Multimedia*, pp. 7434–7443, 2024.

Tongshun Zhang, Pingping Liu, Yubing Lu, Mengen Cai, Zijian Zhang, Zhe Zhang, and Qiuzhan Zhou. CWNet: Causal wavelet network for low-light image enhancement. In *IEEE/CVF International Conference on Computer Vision*, pp. 8789–8799, 2025.

Lei Zhu, Fangyun Wei, Yanye Lu, and Dong Chen. Scaling the codebook size of vqgan to 100,000 with a utilization rate of 99%. In *Advances in Neural Information Processing Systems*, volume 37, 2024a.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *Forty-first International Conference on Machine Learning*, 2024b.

Wenbin Zou, Hongxia Gao, Weipeng Yang, and Tongtong Liu. Wave-mamba: Wavelet state space model for ultra-high-definition low-light image enhancement. In *ACM International Conference on Multimedia*, 2024.

A Appendix

A.1 t-SNE for Group-level Reordering

Motivation. After token-level grouping, tokens assigned to the same codebook entry are gathered into a content group. However, the indices of the learned codebook entries serve only as identifiers and do not encode their relationships in the embedding space. Directly concatenating the groups according to these indices therefore introduces an arbitrary ordering among content groups. For state-space sequence modeling, we instead seek an ordering in which groups associated with related codebook entries tend to occur nearby in the resulting sequence.

To obtain such an ordering, we employ t-SNE (van der Maaten & Hinton, 2008) to organize the learned codebook entries in a low-dimensional space. t-SNE is designed to emphasize local neighborhood structure in high-dimensional representations, making it suitable for identifying a content-aware organization of the codebook entries. Importantly, our objective is not to recover their exact pairwise distances or global geometry. Rather, we use the projected organization only to derive a neighborhood-informed ordering of the content groups.

Relation to Prior Uses of t-SNE. t-SNE was originally introduced as a nonlinear dimensionality-reduction method for constructing low-dimensional representations that emphasize neighborhood relationships in high-dimensional data. This has made t-SNE particularly useful for inspecting the organization of learned representation spaces, where the objective is often to understand whether related high-dimensional representations remain locally organized after projection.

This use is also common in codebook-based representation learning. AdaCode (Liu et al., 2023) applies t-SNE to learned codebook representations to visualize and compare the organization of different codebooks for image restoration. VQGAN-LC (Zhu et al., 2024a) uses t-SNE projections of learned codebook entries to analyze their distribution and utilization. These studies show examples where t-SNE is applied directly to learned code entries, and demonstrate its usefulness for revealing the organization of high-dimensional codebook representations.

Although t-SNE is most commonly used for visualization and representation analysis, its low-dimensional embeddings have also been used as intermediate representations for subsequent computation. Shen et al. (2013) investigated t-SNE as a manifold embedding for constructing compact binary representations used for image retrieval. These works illustrate that the structure obtained through t-SNE can be used beyond visualization and can serve as an intermediate representation for a subsequent computational procedure.

Our use differs from previous works by exploiting the organization induced by t-SNE for sequence construction. We convert the t-SNE embedding into a fixed permutation of the codebook entries, which determines the order in which the corresponding content groups are concatenated for state-space modeling. The t-SNE coordinates themselves are not provided to the network, and their magnitudes or pairwise distances are not used quantitatively. This design does not require the low-dimensional embedding to faithfully reconstruct the complete geometry of the original codebook space. The local neighborhood organization emphasized by t-SNE instead provides a basis for linearizing the unordered high-dimensional codebook entries into a structured sequence of content groups. In our method, t-SNE therefore serves as an intermediate step for deriving the group-level ordering rather than a feature representation directly used by the enhancement network.

Group Ordering. Given the learned codebook $\mathcal{C} = \{\mathbf{e}_k\}_{k=1}^{K}$, we first ℓ_2-normalize each entry, consistent with the normalized embedding space used during codebook learning:

$$\bar{\mathbf{e}}_k = \frac{\mathbf{e}_k}{\|\mathbf{e}_k\|_2}. \tag{15}$$

We then apply one-dimensional t-SNE to the normalized codebook entries:

$$(r_1, \ldots, r_K) = \text{t-SNE}_{\text{1D}}(\bar{\mathbf{e}}_1, \ldots, \bar{\mathbf{e}}_K), \tag{16}$$

where $r_k \in \mathbb{R}$ denotes the projected coordinate of the k-th codebook entry. The group ordering is obtained by sorting these coordinates:

$$\pi = \text{argsort}(r_1, \ldots, r_K). \tag{17}$$

The corresponding content groups are then concatenated according to π:

$$\mathcal{S} = \text{Concat}(\mathcal{G}_{\pi_1}, \mathcal{G}_{\pi_2}, \ldots, \mathcal{G}_{\pi_K}). \tag{18}$$

This procedure converts the neighborhood organization induced by t-SNE into a deterministic one-dimensional ordering that can be directly used for state-space sequence modeling.

t-SNE Settings. We apply t-SNE to the $K = 1024$ codebook entries, each with an embedding dimension of $D = 128$, without additional dimensionality reduction beforehand. The normalized entries are directly embedded into a one-dimensional space with a perplexity of 30. We run the optimization for 1,000 iterations with a fixed random seed of 0 for reproducibility. The resulting permutation π is precomputed once after codebook training and remains fixed throughout LLIE training and inference.

A.2 Codebook Training Details

Training Data and Preprocessing. We train the low-frequency codebook using only the clean target images from the training splits of five UHD image restoration datasets: UHD-LL (Li et al., 2023), UHD-Haze, UHD-Blur (Wang et al., 2024), UHD-Rain, and UHD-Snow (Wang et al., 2026). The numbers of training images used from each dataset are summarized in Table 7.

During training, we randomly crop a 256×256 patch from each selected target image and apply random horizontal flipping for data augmentation. Following the preprocessing described in Sec. 3.2, each patch is decomposed using a single-level 2D Haar DWT (Mallat, 1989), from which only the 128×128 low-frequency (LL) component is retained for codebook learning. The LL component is normalized using the channel-wise statistics computed over the training data according to Eq. 2, with $\epsilon = 10^{-8}$.

Table 7: Training data used for low-frequency codebook learning.

Dataset	Training Images
UHD-LL	2,000
UHD-Haze	2,290
UHD-Blur	1,964
UHD-Rain	3,000
UHD-Snow	3,000
Total	12,254

VQ-VAE Architecture. The VQ-VAE encoder consists of two 4×4 stride-2 convolutional layers with 64 and 128 output channels, respectively, followed by a 3×3 refinement layer and a 1×1 projection layer. Batch normalization and LeakyReLU with a negative slope of 0.1 are applied after each downsampling layer. Given a 128×128 LL component, the encoder produces a 32×32 latent representation $\mathbf{Z}$ with an embedding dimension of $D = 128$.

The latent vectors are quantized using a codebook $\mathcal{C} = \{\mathbf{e}_k\}_{k=1}^{K}$ containing $K = 1024$ entries, where $\mathbf{e}_k \in \mathbb{R}^{128}$. The decoder symmetrically reconstructs the normalized LL component using a 3×3 convolution, two 4×4 stride-2 transposed convolutions, and a final 3×3 convolution followed by a Tanh activation.

Code Assignment and Initialization. The codebook entries are initialized independently from a standard normal distribution,

Code assignment follows the normalized nearest-entry matching described in Sec. 3.2. Given an encoder latent $\mathbf{z}_i$, both the latent and codebook entries are ℓ_2-normalized before computing their distances:

$$q_i = \arg\min_k \left\| \frac{\mathbf{z}_i}{\|\mathbf{z}_i\|_2} - \frac{\mathbf{e}_k}{\|\mathbf{e}_k\|_2} \right\|_2^2, \tag{19}$$

Since both vectors have unit norm, this assignment is equivalent to selecting the codebook entry with the highest cosine similarity. The normalization is used only for determining the assignment; the original unnormalized encoder latents are retained for subsequent codebook updates.

Codebook Update. Rather than optimizing the codebook entries through gradient descent, we update them using exponential moving averages (EMA) of their assigned encoder latents (Van Den Oord et al., 2017). For each codebook entry k, we maintain an EMA assignment count N_k and an EMA latent accumulator $\mathbf{a}_k$. These statistics are initialized as

$$N_k^{(0)} = 0, \qquad \mathbf{a}_k^{(0)} = \mathbf{e}_k^{(0)}. \tag{20}$$

At training step t, we first compute the minibatch assignment count $n_k^{(t)}$ and the sum $\hat{\mathbf{a}}_k^{(t)}$ of the original unnormalized encoder latents assigned to entry k:

$$n_k^{(t)} = \sum_i \mathbb{I}[q_i = k], \qquad \hat{\mathbf{a}}_k^{(t)} = \sum_i \mathbb{I}[q_i = k]\mathbf{z}_i, \tag{21}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function. The EMA statistics are then updated as

$$N_k^{(t)} = \gamma N_k^{(t-1)} + (1-\gamma) n_k^{(t)}, \tag{22}$$

$$\mathbf{a}_k^{(t)} = \gamma \mathbf{a}_k^{(t-1)} + (1-\gamma) \hat{\mathbf{a}}_k^{(t)}, \tag{23}$$

where the EMA decay factor is set to $\gamma = 0.99$.

To avoid unstable updates for rarely assigned entries, we apply Laplace smoothing to the EMA assignment counts:

$$\widetilde{N}_k^{(t)} = \frac{N_k^{(t)} + \epsilon_{\mathrm{EMA}}}{\sum_{j=1}^{K} N_j^{(t)} + K\epsilon_{\mathrm{EMA}}} \sum_{j=1}^{K} N_j^{(t)}, \qquad \epsilon_{\mathrm{EMA}} = 10^{-5}. \tag{24}$$

The corresponding codebook entry is then updated as

$$\mathbf{e}_k^{(t)} = \frac{\mathbf{a}_k^{(t)}}{\widetilde{N}_k^{(t)}}. \tag{25}$$

The normalized representations are used to determine code assignments, whereas the codebook entries are updated from the original unnormalized encoder latents. The codebook is updated entirely through this EMA procedure and receives no gradient-based updates during VQ-VAE training.

VQ-VAE Optimization. While the codebook is updated through EMA, the encoder and decoder are optimized by gradient descent using the reconstruction and commitment objective defined in Eq. 5:

$$\mathcal{L}_{LL} = \left\| \widehat{\mathbf{X}}_{LL} - \widetilde{\mathbf{X}}_{LL} \right\|_2^2 + \beta \left\| \mathrm{sg}[\mathbf{Z}_q] - \mathbf{Z} \right\|_2^2, \tag{26}$$

where $\beta = 0.25$ and $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator. Since the quantization operation is non-differentiable, we use the straight-through estimator to propagate gradients through it:

$$\mathbf{Z}_{\mathrm{ST}} = \mathbf{Z} + \mathrm{sg}[\mathbf{Z}_q - \mathbf{Z}]. \tag{27}$$

Accordingly, the encoder and decoder are optimized through backpropagation, while the codebook entries are updated through EMA. No additional perceptual or frequency-domain losses are used.

Training and Model Selection. We optimize the encoder and decoder using Adam with a constant learning rate of 1×10^{-4}, $(\beta_1, \beta_2) = (0.9, 0.999)$, no weight decay, and a batch size of 16. We train the VQ-VAE for 700 epochs, corresponding to approximately 483K optimization steps, and clip the gradient norm to 1.0. The collected images are randomly split at the image level into 90% and 10% subsets for training and validation, respectively. We select the model according to reconstruction PSNR on the validation subset and use the codebook from the best validation checkpoint in all subsequent experiments.

Codebook Deployment. The VQ-VAE is trained independently before the LLIE network. After codebook learning, the encoder and decoder are discarded, and only the learned codebook $\mathcal{C} \in \mathbb{R}^{1024 \times 128}$ is retained as the set of content anchors used by CASSM. The codebook remains fixed throughout subsequent LLIE training and inference and receives no gradient updates from the enhancement network.

A.3 Effect of CASSM Insertion Position

To examine the effect of applying CASSM at different decoder levels, we consider decoder Level-3 (L3), Level-2 (L2), and Level-1 (L1), ordered from the deepest, lowest-resolution stage to the shallowest, highest-resolution stage. We compare three configurations by progressively inserting CASSM from L3 to L1. All configurations follow the same training and evaluation settings as the ablation experiments in Sec. 4.3.

As shown in Table 8, inserting CASSM at L3 alone achieves 38.99 dB PSNR with 3.95M parameters and 0.449G FLOPs. Extending CASSM to L2 does not provide additional improvement, while increasing both model size and computational cost. Applying CASSM at all three decoder levels yields the highest reconstruction quality, reaching 39.61 dB PSNR and 0.991 SSIM. However, this improvement comes with substantially higher complexity, increasing the parameter count from 3.95M to 6.13M and FLOPs from 0.449G to 2.316G compared with the L3-only configuration. These results indicate that additional CASSM modules do not consistently improve reconstruction quality and introduce considerable computational overhead. We therefore adopt the L3-only configuration as the default design to maintain a favorable balance between enhancement performance and computational efficiency.

A.4 More Qualitative Results

Figures 6 and 7 provide additional qualitative comparisons on UHD-LL and UHD-LOL4K, respectively. The results further demonstrate the consistent enhancement performance of CASSM across

Table 8: Effect of CASSM insertion position.

L3	L2	L1	PSNR	SSIM	Params (M)	FLOPs (G)
✓	×	×	38.99	0.990	**3.95**	**0.449**
✓	✓	×	38.46	0.989	5.04	0.864
✓	✓	✓	**39.61**	**0.991**	6.13	2.316

diverse scenes and image contents. For UHD-LOL4K, where the enhanced results of different methods are visually similar, we additionally visualize the residual maps with respect to the ground truth to facilitate comparison of subtle reconstruction differences.

A.5 AI Use Statement

We use large language models (LLMs) to assist with manuscript writing, including language polishing, grammatical correction, sentence restructuring, and improving the clarity of presentation. All core research ideas, methodological designs, experimental settings, results, analyses, and conclusions are developed and conducted by the authors. All LLM-assisted content is reviewed and verified by the authors.

A.6 Reproducibility Statement

We provide the implementation and experimental details required to reproduce this work in the main paper and supplementary material, including the model architecture, codebook construction and training, optimization settings, datasets, and evaluation protocols. Upon acceptance, we will publicly release the complete source code and configurations for reproducing the experiments presented in this work.

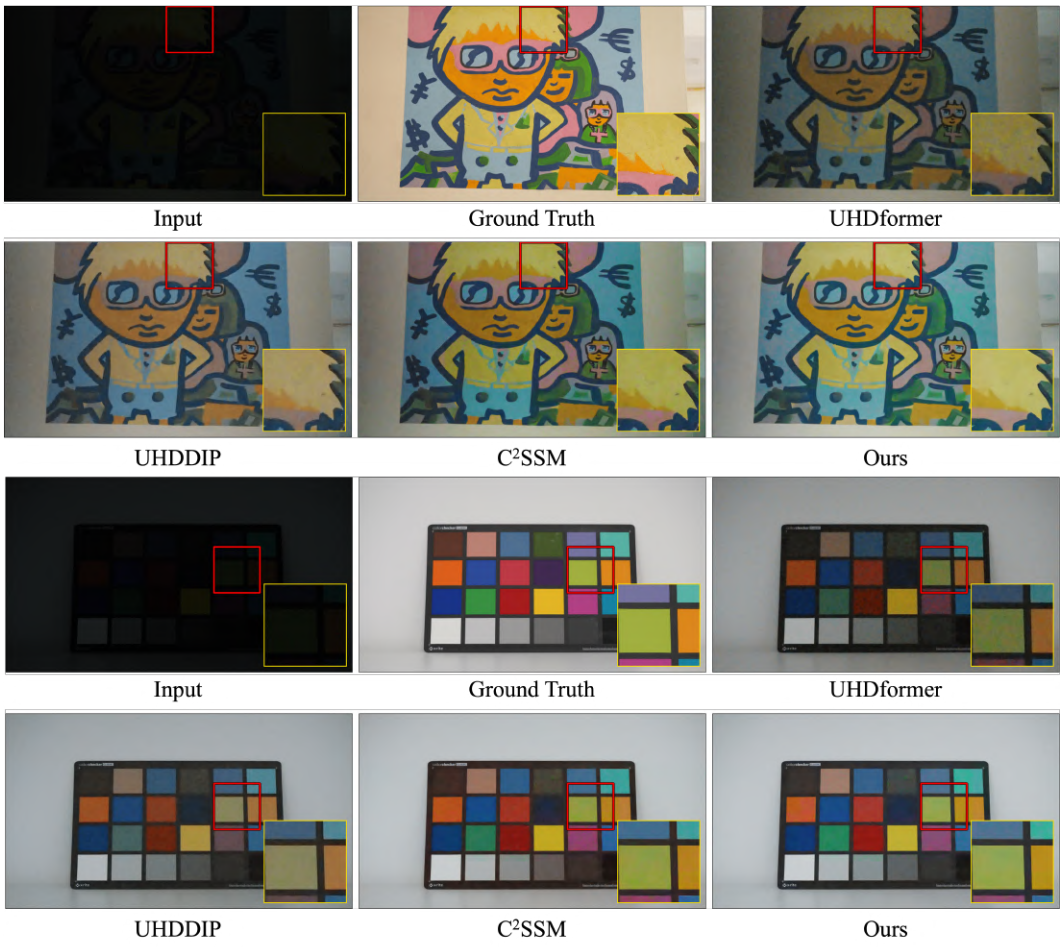


Figure 6: Visual comparison between CASSM and state-of-the-art UHD LLIE methods on UHD-LL dataset.

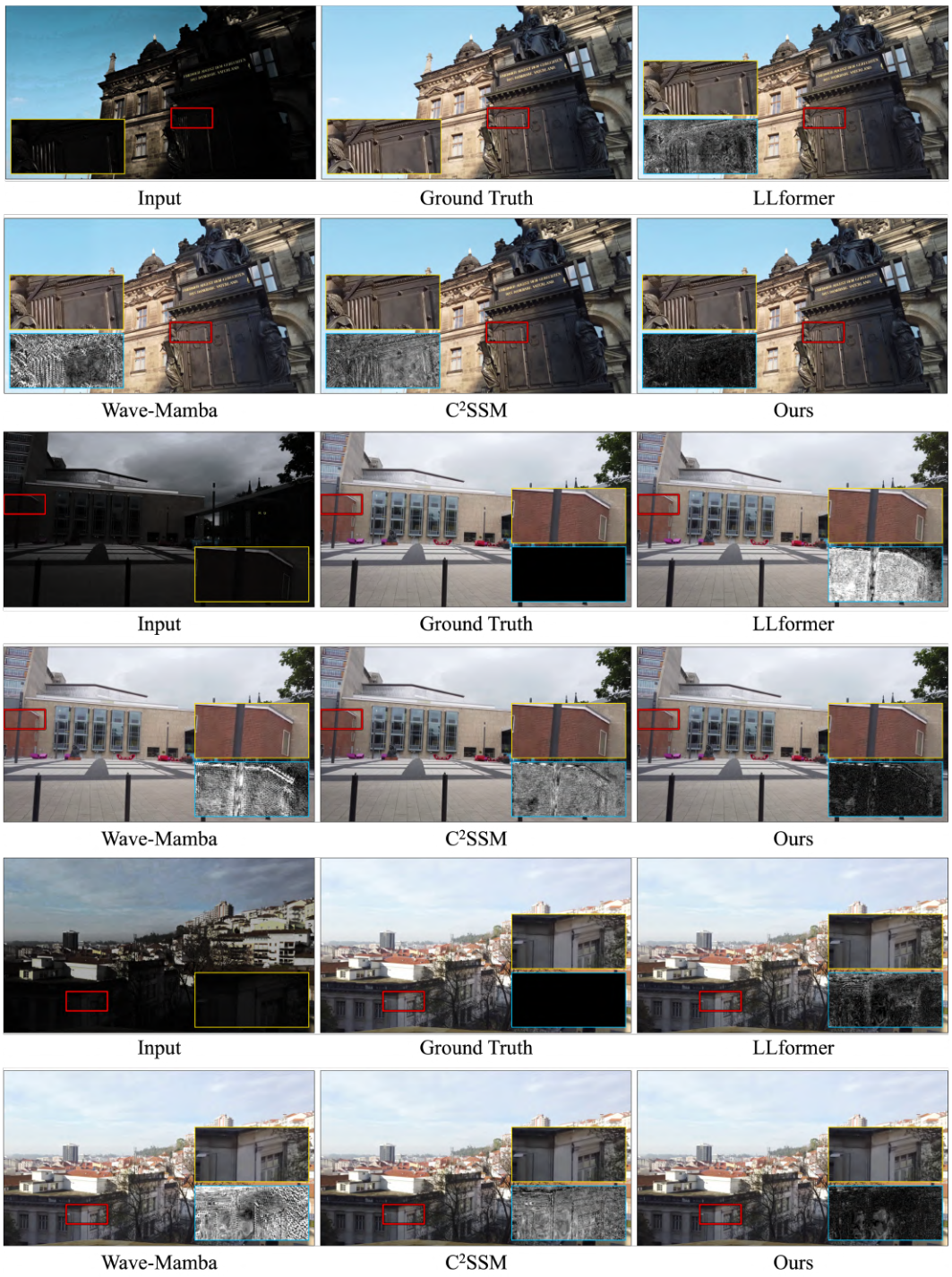


Figure 7: Visual comparison between CASSM and state-of-the-art UHD LLIE methods on the UHD-LOL4K dataset. The red boxes indicate selected regions for detailed comparison. The corresponding enlarged regions and residual maps with respect to the ground truth are highlighted by the yellow and blue boxes, respectively.