AnimateQR: Bridging Aesthetics and Functionality in Dynamic QR Code Generation

Guangyang Wu, Huayu Zheng, Siqi Luo, Guangtao Zhai, Xiaohong Liu*
Shanghai Jiao Tong University
{wu.guang.young, zhenghuayu, siqiluo647,
zhaiguangtao, xiaohongliu}@sjtu.edu.cn

Abstract

Animated QR codes present an exciting frontier for dynamic content delivery and digital interaction. However, despite their potential, there has been no prior work focusing on the generation of animated QR codes that are both visually appealing and universally scannable. In this paper, we introduce AnimateQR, **the first generative framework** for creating **animated QR codes** that balance aesthetic flexibility with scannability. Unlike previous methods that focus on static QR codes, AnimateQR leverages **hierarchical luminance guidance** and **progressive spatiotemporal control** to produce high-quality dynamic QR codes. Our first innovation is a multi-scale hierarchical control signal that adjusts luminance across different spatial scales, ensuring that the QR code remains decodable while allowing for artistic expression. The second innovation is a progressive control mechanism that dynamically adjusts spatiotemporal guidance throughout the diffusion denoising steps, enabling fine-grained balance between visual quality and scannability. Extensive experimental results demonstrate that AnimateQR achieves state-of-the-art performance in both decoding success rates (96% vs. 56% baseline) and visual quality (user preference: 7.2 vs. 2.3 on a 10-point scale). Codes are availble at https://github.com/mulns/AnimateQR.

1 Introduction

Aesthetic QR codes have emerged as a promising medium that integrates machine-readable functionality with human-oriented visual design [41, 31, 22, 14]. Early approaches typically focused on module deformation [3, 10, 45, 4, 16] and style transfer [41, 41] to improve visual appeal. With the advancement of generative models [32, 36, 15, 35, 17, 19, 18, 20, 26, 28, 27], recent methods employ ControlNet-based frameworks [6, 43] to synthesize stylized QR code images [34]. To further ensure scanning reliability, dedicated mechanisms [22, 34, 39, 5] have been introduced to balance the trade-off between aesthetics and robustness.

Unlike existing static QR codes, animated QR codes encoded as video sequences offer enhanced branding potential, interactive storytelling, and context-aware content delivery. However, generating temporally coherent animated QR codes that maintain robust scannability while achieving artistic expressiveness presents significant challenges.

To the best of our knowledge, we are the first to tackle animated QR code generation, addressing a critical gap in existing methods that are primarily designed for static image domains. Directly applying these static methods [4, 3, 22, 34, 39] to animated scenario in a frame-by-frame manner results in visually unappealing outcomes, often producing rigid and unnatural animations due to a lack of temporal coherence and aesthetic consistency. Our motivation arises from the observation that overly fine-grained control leads to instability, whereas overly coarse control introduces visual

* Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

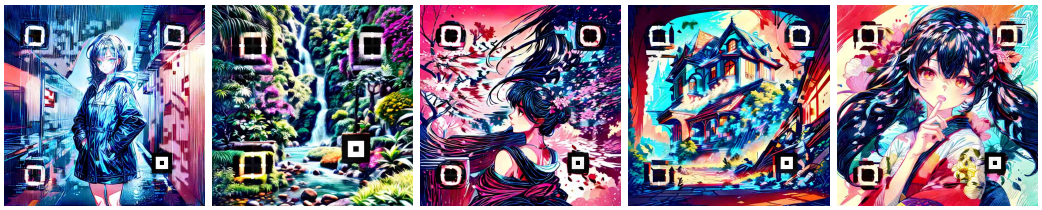

Figure 1: Samples of animated QR codes are shown. *Best viewed with Acrobat Reader. Click the images to play the animation clips and zoom in for finer details.*

artifacts. To balance robustness and visual quality, we propose a novel framework with adaptive control granularity, *enabling dynamic adjustment of spatial and temporal constraints to prioritize scannability in critical regions while allowing creative freedom in less important areas*. We name the method AnimateQR, which introduces two key innovations:

1. Hierarchical Luminance Guidance (HLG). HLG is a multi-scale luminance map that can be derived from both natural images and QR codes. It encodes hierarchical structural information, serving as a control mechanism to bridge artistic expression and QR code functionality. Our proposed HLG-ControlNet, trained on natural images and their corresponding HLG maps, achieves superior performance than existing luminance ControlNet [43] through three key innovations:

(1) Module-aligned Encoding: Spatial partitioning aligns with QR decoder's module-wise processing requirements, ensuring structural compatibility. (2) Hierarchical Control: Multi-scale constraint mechanism enables dynamic granularity adaptation across different QR module hierarchies. (3) Adaptive Constraint: Ternary quantization strategy selectively preserves critical regions while allowing flexibility in non-essential areas through stochastic relaxation.

2. Progressive SpatioTemporal control (ProST). Building upon the HLG-ControlNet framework, we propose a novel diffusion-based progressive spatiotemporal control mechanism that jointly optimizes visual quality and scannability during the inference phase. Unlike conventional approaches that enforce uniform control across frames, ProST dynamically partitions control strength across both spatial and temporal dimensions:

◇ *Temporally*: ProST assigns distinct HLG maps to each frame through a combination of reshuffling and interpolation, ensuring temporal coherence while maintaining scannability.

◇ *Spatially*, the HLG control signals adaptively evolve across diffusion stages: regions prone to decoding errors receive stronger guidance, while more stable areas transition to softer constraints.

This dual-domain adaptation mechanism effectively balances video coherence with decoding reliability, achieving an optimal trade-off between visual appeal and functional robustness.

Extensive experimental results demonstrate that AnimateQR achieves state-of-the-art performance in both decoding success rates (96% vs. 56% baseline) and visual quality (user preference: 7.2 vs. 2.3 on a 10-point scale).

The key contributions of this paper are threefold: 1) First generative framework for animated QR codes. 2) Hierarchical Luminance Guidance (HLG) that enhances control granularity spatially. 3) Progressive Spatiotemporal Control (ProST): a diffusion-based method for error-adaptive control across space and time.

2 Related Work

Quick Response (QR) Code. As QR codes increasingly serve as a vital bridge between physical and digital domains, there is growing interest in enhancing their visual aesthetics beyond the traditional monochrome design. Early efforts, such as halftone QR codes [3], align QR modules with thematic images to create visually integrated patterns. Artup [7, 40] further explore the embedding of colorful content within QR structures. For module rearrangement, the QArt team [4, 16] pioneered the application of Gaussian-Jordan elimination algorithms to spatially reorganize encoding units. Subsequent advances [30, 31] apply artistic style transfer techniques to enrich QR code appearance.

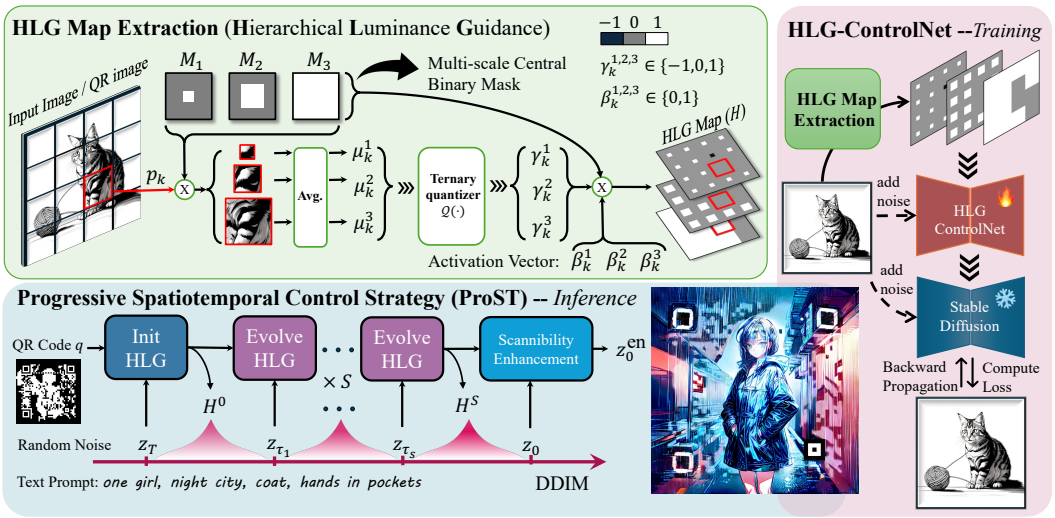

Figure 2: Illustration of the proposed AnimateQR framework. The green box depicts the extraction of the Hierarchical Luminance Guidance (HLG) map from input images. The red box outlines the training of the HLG-ControlNet, while the blue box highlights the inference stage using the proposed Progressive Spatiotemporal control (ProST) strategy. During training, we extract HLG map with randomly sampled activation vector β_k as detailed in Section 3.3. During inference, animated QR codes are generated via the DDIM process using a Stable Diffusion model equipped with HLG-ControlNet (as the control unit) and AnimateDiff (as the motion module). The HLG map is progressively updated by the Evolve HLG module as detailed in Section 3.4, enabling adaptive spatiotemporal control throughout the denoising process.

Style transfer technology further elevates the artistic quality of QR codes. Xu et al. [41] proposed SEE QR codes, marking the advent of aesthetically pleasing QR codes based on style transfer. Subsequently, Su et al. [30, 31] enhanced the aesthetic effect through a module-based deformable convolution mechanism (MDCM), generating QR codes that are both recognizable by ordinary mobile scanners and artistically expressive.

Diffusion for Aesthetic QR Image: In recent years, deep learning–based image manipulation and generation techniques have made significant progress [32, 36, 15, 35, 17, 19, 18, 20], with generative models leading the forefront of this advancement [42, 21, 24, 23, 26, 28, 27]. QRBTF [22] employs diffusion models [25] to generate photorealistic QR codes with natural textures while maintaining robust scannability. Meanwhile, GladCoder [39] integrates GANs [8] and diffusion models [25] to refine high-frequency details in QR codes, enabling them to mimic complex artworks without sacrificing machine readability, thus solving the problem of artistic expression limitations in traditional QR designs. The QRMonster model [6], built upon the ControlNet framework, enhances structural control capabilities of diffusion models through its adapter module [46]. Text2QR [34] leverages QRMonster to improve visual output quality while optimizing positioning patterns via a three-stage refinement process, thereby significantly enhancing the overall aesthetic quality of QR code images. Face2QR [5] represents another noteworthy research direction, which explores the integration of facial features into QR code design to create unique and highly personalized solutions. This innovation undoubtedly opens up new possibilities for the future of QR code aesthetics.

3 Method

3.1 Preliminary

A QR code is decoded through three primary steps:

1. *Detection and Localization*: Locate the QR code region by detecting Finder Patterns (the square markers at three corners) and Alignment Patterns, and determine the QR code's version and format information.

2. *Module Sampling and Binarization*: Partition the QR code image into a grid of $m \times m$ modules, where each module is assigned a binary value (0 or 1) based on its luminance level.

3. *Error Correction and Decoding*: Decode the binary sequence using Reed-Solomon error correction to retrieve the original message.

In this paper, we define all QR code images as composed of $m \times m$ modules, where each module consists of $a \times a$ pixels. For convenience, we standardize all images to a size of $ma \times ma$.

3.2 Overall Framework

The overall framework of our approach is illustrated in Figure 2. During inference, our method follows the standard Denoising Diffusion Implicit Models (DDIM) [29] process to denoise a randomly sampled noise using Stable Diffusion (v1.5) [25]. Additionally, we employ ControlNet [43] for spatial luminance control and AnimateDiff [9] for motion generation in video sequences. Specifically, the ControlNet we utilize adheres to the standard model architecture but is trained on a novel Hierarchical Luminance Guidance (HLG) signal dataset, referred to as **HLG-ControlNet**. During the denoising process, we dynamically adjust the control signals through a mechanism named **Progressive Spatiotemporal Control (ProST)**, assigning varying control strengths both spatially and temporally to achieve a fine-grained balance between visual quality and scanning robustness.

The rationale for selecting AnimateDiff as the motion module can be summarized as follows: (1) AnimateDiff is seamlessly integrated with the image generation model Stable Diffusion, which benefits from a rich ecosystem of community fine-tuned stylized models. (2) By leveraging AnimateDiff, we can apply control through image-based ControlNet without the need to train a separate video ControlNet. This design makes our method highly extensible, enabling both static and animated QR code generation with minimal additional effort. (3) Unlike most video generation models, AnimateDiff supports frame-wise luminance control, which naturally aligns with the requirements of animated QR code generation.

In the subsequent sections, we elaborate on the training process of HLG-ControlNet and the inference process of ProST for animated QR code generation.

3.3 Training HLG-ControlNet

As previously mentioned, although our framework generates animated QR codes in video sequences, we leverage image-based ControlNet to achieve precise luminance control. As illustrated in the red box of Fig. 2, we train the ControlNet following the standard process [43] on a dataset of images paired with their corresponding Hierarchical Luminance Guidance (HLG) maps. The network is trained end-to-end, with HLG maps provided as conditional inputs. We adopt the same loss functions and optimization strategy as ControlNet to ensure reproducibility while focusing on learning HLG-guided control signals.

The HLG map is defined as a three-channel representation, where each channel corresponds to a distinct scale. Each pixel in the HLG map assumes one of three values: -1 for dark regions, 1 for bright regions, and 0 for regions with unrestricted luminance.

The subsequent section details the process of extracting the HLG map from an input image, as further illustrated in the green box of Figure 2.

Multi-scale Luminance Extraction. For input image $I \in \mathbb{R}^{ma \times ma}$, we partition it into m^2 non-overlapping $a \times a$ patches $P = \{p_k\}_{k=1}^{m^2}$ via grid decomposition. Each patch is analyzed using three *Multi-scale Central Masks* $\{M_r \in \mathbb{R}^{a \times a}\}_{r=1}^{3}$, defined with central radii $c_r = \lfloor a/2^{r-1} \rfloor$:

$$M_r(x, y) = \begin{cases} 1 & \text{if } (x, y) \in \text{Center}(c_r \times c_r), \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The hierarchical luminance values are calculated through:

$$\mu_k^r = \frac{1}{\|M_r\|_0} \sum_{(x,y) \in \Omega_k} p_k(x, y) \cdot M_r(x, y), \tag{2}$$

where Ω_k is the pixel coordinate space within patch p_k and $r \in \{1, 2, 3\}$ denotes three scales.

Hierachical Luminance Guidance. Given the average luminance values across different scales, we quantize them to align with the distribution of QR code images. We define a ternary quantization function $\mathcal{Q} : [0, 1] \to \{-1, 0, 1\}$ as follows:

$$\mathcal{Q}(\mu) = \begin{cases} -1 \text{ (Dark)} & \text{if } \mu < \tau_b, \\ 1 \text{ (Bright)} & \text{if } \mu > \tau_a, \\ 0 \text{ (Unconstrained)} & \text{otherwise,} \end{cases} \tag{3}$$

where $\tau_a = 0.7$ and $\tau_b = 0.3$ are decoder-calibrated thresholds derived from sensitivity analysis. For each patch p_k, the average luminance μ_k^r is mapped to the ternary luminance vector $\gamma_k = \left[\mathcal{Q}(\mu_k^1),\ \mathcal{Q}(\mu_k^2),\ \mathcal{Q}(\mu_k^3)\right]$ according to Equation 2 - 3.

To dynamically activate scale-specific constraints, we introduce an **activation vector** $\beta_k \in \mathcal{B}$. Here, $\mathcal{B}$ is explicitly defined as the **ordered basis set**:

$\mathcal{B} \triangleq (b_1, ..., b_8) = \{(0, 0, 0), (0, 0, 1), (0, 1, 0), (1, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 0), (1, 1, 1)\}$. Each basis b_i (for $i = 1, 2, \ldots, 8$) controls the activation of the three scales, with 0 indicating deactivation and 1 indicating activation, thereby enumerating all possible activation patterns. Each value in b_i corresponds to one scale: activating a scale enables its corresponding blocks to contribute to the content control strength. For instance, (1, 0, 0) modifies only coarse-scale blocks, resulting in weaker control, whereas (1, 1, 1) involves all scales and achieves maximal control strength. The definition of $\mathcal{B}$ and b_i facilitates the explanation of how the HLG map is updated by modifying the activation vector during inference, as detailed in the following section.

The final HLG map H is constructed by combining multi-scale constraints through channel-wise concatenation (denoted as $\bigoplus$):

$$H_k = \bigoplus_{r=1}^{3} \left[\gamma_k^r \cdot M_r \cdot \beta_k^r\right], \quad H = \text{unpatchify}(\{H_k\}_{k=1}^{m^2}). \tag{4}$$

During training, we uniformly explore all patterns by sampling β_k such that $P(\beta_k = b_i) = \frac{1}{8}$ for all $i \in \{1, \ldots, 8\}$. This ensures that the HLG-ControlNet becomes robust to spatially dynamic control strengths.

3.4 Inference with ProST

Our diffusion-based Progressive Spatiotemporal Control (ProST) framework addresses the long-standing control-strength dilemma through an *error-adaptive constraint escalation* mechanism. Specifically, ProST dynamically modulates hierarchical constraints in response to error-driven control escalation, operating through three distinct phases: *Init HLG*, *Evolve HLG*, and *Scannability Enhancement*. This phased approach ensures a systematic and adaptive progression in constraint enforcement, significantly enhancing the framework's robustness and precision in spatiotemporal control tasks. The complete algorithm is presented in Algorithm 1.

Notation. Let $k \in \{1, \ldots, m^2\}$ index the QR modules, where m^2 represents the total number of modules in the QR code. Let $\{z_{\tau_s}\}_{s=0}^{S}$ denote the latent states at evenly spaced timesteps, where $\tau_s = T - s\Delta_\tau$ and $\Delta_\tau = T/S$. At each stage s, conditional denoising is performed as:

$$z_{\tau_{s+1}} = \text{DDIM}(z_{\tau_s}, H^s, \tau_s \to \tau_{s+1}), \tag{5}$$

where H^s is the adaptive HLG map, dynamically updated based on intermediate decoding results.

Given an animated QR sequence V consisting of N frames, we partition it into two disjoint subsets: the keyframe set $\mathcal{K} \subset \{1, 2, \ldots, N\}$ and the non-keyframe set $\mathcal{T} = \{1, 2, \ldots, N\} \backslash \mathcal{K}$. The keyframes $\mathcal{K}$ are uniformly sampled and are required to satisfy scannability constraints, while the non-keyframes $\mathcal{T}$ primarily focus on maintaining temporal coherence without enforcing scannability.

Init HLG. Given a QR code q and random noise z_T, this module generate a initialized HLG map H^0. First, the initial video sequence $V' = \{I_t\}_{t=1}^{N}$ is generated as:

$$V' = \mathcal{D}_{\text{vae}}\left(\text{DDIM}(z_T, H^{\text{null}}, T \to 0)\right), \tag{6}$$

where H^{null} denotes null control, N represents the number of video frames (typically set to 16), and $\mathcal{D}_{\text{vae}}$ is the VAE Decoder [13].

Next, we align the QR modules with the video content through the following operations:

$$q'_t = g(V', q) = \begin{cases} \text{Reshuffle}(I_t, q), & \text{if } t \in \mathcal{K}; \\ \text{Interp}(q'_{t^-}, q'_{t^+}, \alpha_t), & \text{if } t \in \mathcal{T}; \end{cases} \tag{7}$$

where: Reshuffle$(\cdot,\cdot)$ is derived from QArt [4] (details provided in the Appendix); Interp$(\cdot)$ performs nearest-neighbor interpolation between two keyframes based on temporal distance; $t^- = \max\{i \in \mathcal{K} \mid i < t\}$ and $t^+ = \min\{i \in \mathcal{K} \mid i > t\}$ denote the nearest preceding and succeeding keyframes relative to frame t; $\alpha_t = \frac{t-t^-}{t^+-t^-} \in (0,1)$ represents the temporal interpolation ratio; $q'_t \in \mathbb{R}^{m^2}$ is the reshuffled binary sequence with $q'_t[k] \in \{-1, 1\}$.

Subsequently, we construct the HLG map for each frame in parallel, where the temporal subscript is omitted for simplicity. According to Eq. 4, the HLG map H^s at stage s is formulated as:

$$H^s = f(q', \{\beta_k^{(s)}\}_{k=1}^{m^2}) = \text{unpatchify}\left(\left\{\bigoplus_{r=1}^{3}[q'[k] \cdot M_r \cdot \beta_k^{r(s)}]\right\}_{k=1}^{m^2}\right). \tag{8}$$

At the initial stage $s = 0$, we set $\beta_k^{(0)} = b_2$ as the initialized activation state.

Evolve HLG. At stage $s > 0$, given the current estimate z_{τ_s} from Eq.5 and the reshuffled QR code q', the module updates the HLG map from H^s to H^{s+1}. As defined in Eq.8, this is essentially achieved by updating the activation vector set $\{\beta_k^{(s)}\}_{k=1}^{m^2}$. To achieve error-driven control updates, we first analyze the error distribution. Specifically, the current estimate z_{τ_s} is decoded as $\hat{q}^s = \mathcal{D}_{\text{qr}}(\mathcal{D}_{\text{vae}}(z_{\tau_s}))$, where $\mathcal{D}_{\text{qr}}$ denotes the simulated QR decoder [1] (details provided in the Appendix). The module-wise error mask is then computed as:

$$E_k^s = \mathbb{I}[\hat{q}^s[k] \neq q'[k]], \quad 1 \leq k \leq m^2, \tag{9}$$

where $\mathbb{I}[\cdot]$ is the indicator function, returning 1 if the condition is true and 0 otherwise.

Next, we define the evolve function as follows:

$$\beta_k^{(s+1)} = \begin{cases} b_{\min(\phi(\beta_k^s)+1,8)}, & \text{if } E_k^s = 1; \\ b_{\max(\phi(\beta_k^s)-1,1)}, & \text{otherwise}, \end{cases} \tag{10}$$

where $\phi : \mathcal{B} \to \{1, \ldots, 8\}$ is a bijective index mapping satisfying $\phi(b_i) = i$. After this, the updated HLG map H^{s+1} is computed following Eq. 8. This update rule ensures that *for erroneous modules, the control strength is increased, while for correct modules, the control strength is decreased*, thereby achieving a balanced trade-off between scannability and aesthetic quality.

Scannability Enhancement. After the final denoising step, the latent code is refined using the SELR (Scannability Enhancement via Latent Refinement) module [34]. This module applies gradient-based iterative refinement as follows:

$$z_0^{\text{en}} = z_0 - \eta \sum_{i=1}^{L} \nabla_{z_0} \mathcal{L}_{\text{scan}}\left(q', \mathcal{D}_{\text{qr}}(\mathcal{D}_{\text{vae}}(z_0^{(i)}))\right), \tag{11}$$

where: L is the number of SELR iterations, $z_0^{(i+1)} = z_0^{(i)} - \eta \nabla \mathcal{L}_{\text{scan}}$ represents the iterative update with learning rate η, $\mathcal{L}_{\text{scan}}$ is the scannability loss function (detailed in the Appendix). This refinement process ensures that the final latent code z_0^{en} is optimized for scannability while preserving the visual quality of the generated QR code.

4 Experiment

Algorithm 1 Progressive Spatiotemporal Control (ProST)

Require: Initial noise z_T, target QR q, stages S
Ensure: Scannable V^{qr}

Initialize: $V' \leftarrow \mathcal{D}_{\text{vae}}\left(\text{DDIM}(z_T, H^{\text{null}}, T \to 0)\right)$
Let $q' \leftarrow g(V', q)$
Let $\beta_k^0 \leftarrow (0, 0, 1)\ , \forall k \in \{1, 2, \ldots, m^2\}$
Let $H^0 \leftarrow f(q', \{\beta_k^{(0)}\}_{k=1}^{m^2})$
Let $z_{\tau_0} \leftarrow z_T$
for $s = 1$ to S **do**
    Denoise: $z_{\tau_s} \leftarrow \text{DDIM}(z_{\tau_{s-1}}, H^{s-1}, \tau_{s-1} \to \tau_s)$
    Decode: $\hat{q}^s \leftarrow \mathcal{D}_{\text{qr}}(\mathcal{D}_{\text{vae}}(z_{\tau_s}))$
    for $k = 1$ to m^2 **do**
        if $\hat{q}_k^s \neq q'_k$ **then**
            $\beta_k^s \leftarrow b_{\min(\phi(\beta_k^{s-1})+1,8)}$
        else
            $\beta_k^s \leftarrow b_{\max(\phi(\beta_k^{s-1})-1,1)}$
        end if
    end for
    Update HLG: $H^s \leftarrow f(q', \{\beta_k^{(s)}\}_{k=1}^{m^2})$
end for
Enhance Scannability: $z_0^{\text{en}} \leftarrow \text{SELR}(z_0)$
Output: $V^{\text{qr}} \leftarrow \mathcal{D}_{\text{vae}}(z_0^{\text{en}})$

Experimental Setup and Configuration. Our implementation is based on the PyTorch framework and runs on an NVIDIA GeForce RTX 4090 GPU. We generate QR codes of version 5, corresponding to 37×37 modules (i.e., $m = 37$). For training the HLG-ControlNet, we adopt *Stable Diffusion v1.5* as the backbone and utilize a dataset comprising 60,000 high-resolution images, each preprocessed to a resolution of 512×512 pixels. During inference, we set the ControlNet control strength to 0.9, the number of frames in AnimateDiff to 16, and the motion scale to 1.0. By default, we define the keyframe set as $\mathcal{K} = \{1, 8, 16\}$ and set the learning rate to $\eta = 0.001$.

For comparative evaluation, we construct a dataset of 500 uniquely stylized QR images, each with a resolution of 1024×1024 pixels, encompassing diverse visual content and artistic styles. To assess scanning robustness, all generated results are displayed on a 27-inch, 144Hz IPS monitor. The scannability of dynamic QR codes is tested under real-world conditions by playing the 16-frame animation on the screen and performing actual scanning, consistent with their intended practical usage.

As mentioned earlier, our proposed AnimateQR can be easily extended to generate **static** aesthetic QR code images. This is achieved by disabling AnimateDiff and setting $\mathcal{K} = \{1\}$. We refer to this static variant as AnimateQR-*s*.

4.1 Qualitative Comparison

Animated QR Code Quality. Due to the absence of dedicated methods for animated QR code generation, we extend two representative static image-to-QR approaches, ArtCoder [31] and GladCoder [39], to the video domain via frame-by-frame processing, denoted as "ArtCoder-*d*" and "GladCoder-*d*", ensuring fair comparison under identical conditions. As shown in Fig. 3, these baselines exhibit noticeable temporal flickering due to the lack of inter-frame modeling, whereas our method generates animations with significantly improved temporal coherence and visual continuity.

Static QR Code Quality. For fair comparison, we evaluate our static variant, AnimateQR-*s*, against state-of-the-art static QR code generation methods. The comparison includes image-to-QR methods (ArtCoder [31], GladCoder [39]) and text-to-QR methods (QRBTF [22], Text2QR [34]). To accommodate varying input requirements, we use unified prompt-image pairs as dual conditioning inputs (details are in the Appendix). As Figure 4 shows, ArtCoder, QRBTF, and GladCoder suffer from color-block artifacts, while Text2QR improves aesthetics but exhibits unnatural textures and color shifts. In contrast, AnimateQR-s delivers visually coherent QR codes with seamless module integration, thanks to our adaptive control strategy that spatially modulates hierarchical constraints based on reconstruction errors.

4.2 Quantitative Comparison

Scanning Robustness. In this study, we evaluate the scanning robustness of our QR codes in comparison with two methods: Text2QR [34] and QRBTF [22]. We generate a set of 50 aesthetically optimized QR images for each method, all at a resolution of 1024×1024 pixels. These images are displayed on a high-definition monitor at three standard sizes: 3 cm^2, 5 cm^2, and 7 cm^2. During controlled testing, smartphones are held at a fixed distance of 40 cm, and each QR code is scanned

Table 1: Average scanning success rates (%) are assessed across various phone applications, considering different sizes (cm^2) and angles (°). "Scanner" denotes the native scanner of system. We compare our method with QRBTF [22] and Text2QR [34] under same condition.

Decoders	**Success Rate (%)**					
	$(3cm)^2$		$(5cm)^2$		$(7cm)^2$	
	45°	90°	45°	90°	45°	90°
	QRBTF [22]					
Scanner	100	100	100	100	100	100
TikTok	96	96	78	83	56	72
WeChat	100	100	100	98	94	100
	Text2QR [34]					
Scanner	100	100	100	100	100	100
TikTok	100	100	100	100	96	100
WeChat	100	100	96	100	94	100
	AnimateQR					
Scanner	100	100	100	100	100	100
TikTok	100	100	100	100	96	100
WeChat	100	100	100	100	96	96

Table 2: Comparison of animated QR code generation with **best** results in bold.

Methods	**Q-Bench↑**	**SimpleVQA↑**	**Speed↑**
ArtCoder-*d*	0.2954	3.0254	0.031
GladCoder-*d*	0.2453	2.6901	0.016
AnimateQR	**0.6217**	**3.5872**	**0.613**

Table 3: Comparison of static QR code generation

Methods	**Q-Align↑**	**LIQE↑**	**AesBench↑**
ArtCoder	0.6003	2.6363	0.4396
GladCoder	0.6559	3.2529	0.7729
QRBTF	0.7877	3.5447	0.7822
Text2QR	0.7788	3.5034	0.7610
AnimateQR-*s*	**0.8433**	**3.8572**	**0.8832**

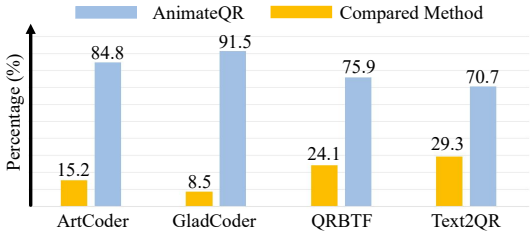

Figure 5: Statistical results of user study.

for 3 seconds from various angles. Scanning success rates, averaged over 20 trials, are reported in Table 1. Our method consistently achieves a success rate above 96%, outperforming the baselines. Notably, QR codes failing within the 3-second window were often successfully scanned with extended exposure, demonstrating the practical reliability of our generated codes.

IQA / VQA Metrics and Inference Speed. To assess static QR code quality, we adopt Q-Align [38] (0 – 1), LIQE [44] (0 – 5), AesBench [12] (0 – 1), and AesExpert [11, 33] (0 – 1) as aesthetic evaluation metrics. For animated QR codes, we employ Q-Bench [37] (0 – 1) and SimpleVQA [2] (0 – 5). In addition, we report the average inference speed (samples per second) for animated QR code generation. All metrics and runtime statistics are averaged over 100 generated samples per method and summarized in Table 2 - 3. Our method consistently achieves the best performance across both static and animated QR code evaluations.

User Study. To further assess the practical effectiveness of our method, we conduct a user study as a supplementary subjective evaluation. The user study consisting of 30 participants to evaluate

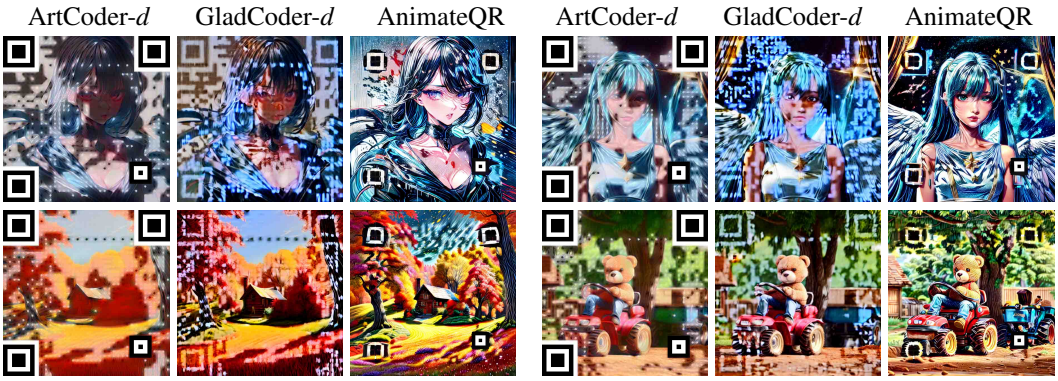

Figure 3: Visual comparison of animated QR code generation methods. *Best viewed with Acrobat Reader. Click the images to play the animation clips and zoom in for finer details.*

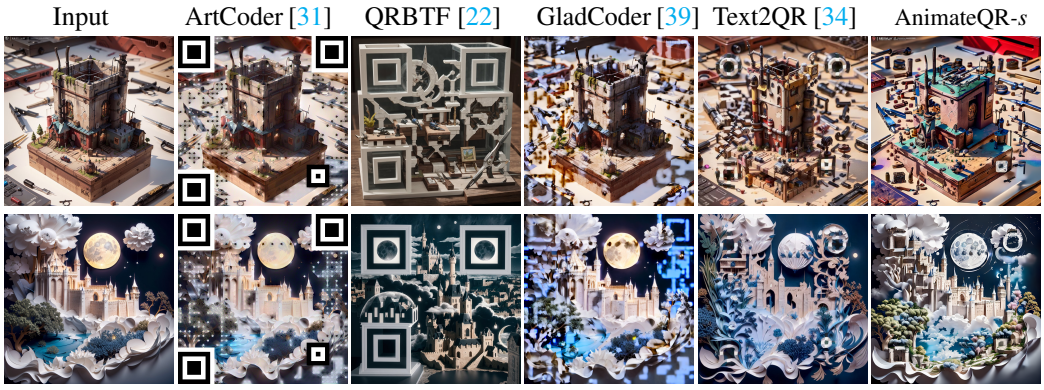

Figure 4: Visual comparison of static QR code generation methods.

Table 4: Average QR code scanning success rates (%) across illumination levels ranging from 500 to 5 LUX.

Methods	500	400	300	200	100	5
Text2QR-*d*	100	100	96	82	64	2
QRBTF-*d*	98	96	90	62	48	0
AnimateQR	100	100	98	88	66	8

Table 5: Average QR code scanning success rates (%) across motion blur levels ranging from 0.1 to 2.0 m/s.

Methods	0.1	0.5	1.0	1.5	2.0
Text2QR-*d*	100	100	96	90	88
QRBTF-*d*	98	98	92	82	70
AnimateQR	100	100	96	92	88

Table 6: Average QR code scanning success rates (%) across scanning angles ranging from $0°$ to $60°$.

Methods	$0°$	$15°$	$30°$	$45°$	$60°$
Text2QR-*d*	100	100	100	100	98
QRBTF-*d*	98	100	98	100	96
AnimateQR	100	100	100	100	98

Table 7: Comparison of AnimateQR variants. AnimateQR-XL uses SDXL + AnimateDiff-XL; AnimateQR-LCM uses AnimateDiff-LCM.

Methods	**Q-Bench↑**	**SimpleVQA↑**
AnimateQR	0.6217	3.5872
AnimateQR-XL	0.6926	3.8216
AnimateQR-LCM	0.6138	3.5921

200 generated QR codes (50 for each methods) generated by different methods (the approval from Institutional Review Board is obtained). Participants are asked to rank the results in the aspect of aesthetic quality. The percentages represent how many users prefer the results of a method over others. The results are illustrated in Figure 5, with detailed experimental settings provided in the Appendix. The results indicate a clear user preference for our method over existing baselines.

Real-World Robustness. To rigorously evaluate the robustness of our method in real-world scenarios, we conduct three types of tests: varying illumination, motion blur, and scanning angles. As benchmarks, we extend Text2QR and QRBTF to dynamic settings via per-frame generation, denoted as "Text2QR-*d*" and "QRBTF-*d*". We perform side-by-side comparisons on 50 QR code sequences (each $5\,\text{cm} \times 5\,\text{cm}$), ensuring a fair and comprehensive evaluation. We conduct controlled experiments: ambient illumination is varied from near 0 to 500LUX across six levels with the phone fixed at $0°$, motion blur is simulated along a 3m path at speeds of 0.1–2m/s under 500LUX, and scanning angles are tested from $0°$ to $60°$ with stationary phones under 500LUX. Each method scans the same QR codes 10 times per condition, and average success rates are recorded. The results, summarized in Table 4–6, show that AnimateQR consistently outperforms Text2QR-*d* and QRBTF-*d* under realistic conditions, including low light, motion blur, and non-standard scanning angles, achieving superior stability and decoding reliability in challenging scenarios.

Model Generality. We evaluate the generality and scalability of our framework across different model variants. As shown in Table 7, AnimateQR-XL (SDXL + AnimateDiff-XL) shows substantial improvements over the SD-v1.5 variant, while AnimateQR-LCM (with the distilled AnimateDiff-LCM) achieves comparable performance. These results demonstrate our framework's extensibility to newer architectures and generalizability to distilled models. Moreover, our framework supports style generality by integrating LoRA-trained diffusion models to modify output styles.

Table 8: HLG-ControlNet Ablation Comparison. Details are in Section 4.3.

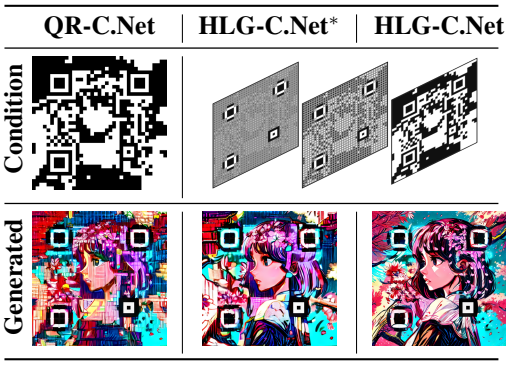

Table 9: ProST Ablation. We compare ProST (adaptive β_k) with static baselines using fixed $\beta_k = b_2$ (light control) and $\beta_k = b_8$ (strong control). ProST achieves better balance between visual quality and scannability, as reflected by scanning error rates.

4.3 Ablation Study

HLG-ControlNet. While prior methods such as Text2QR [34], Face2QR [5], and GladCoder [39] adopt QRMonster [6] as the ControlNet conditioned on QR code images, we propose HLG-ControlNet, which leverages hierarchical luminance guidance for conditioning. To enhance robustness during training, we introduce random sampling of the modulation vector β_k to simulate varying control strengths. We conduct ablation studies by replacing HLG-ControlNet (referred to as HLG-C.Net) in AnimateQR with: (1) QRMonster, denoted as QR-C.Net, which serves as the standard ControlNet conditioned on QR code images, and (2) HLG-C.Net*, a variant trained with a fixed modulation vector $\beta_k = b_8$, thereby disabling adaptive control. As shown in Table 8, QRMonster often introduces blocky artifacts. HLG-C.Net* benefits from hierarchical luminance guidance, yielding more coherent outputs. Our full model achieves the best visual quality, validating the effectiveness of both HLG-based conditioning and adaptive modulation.

Progressive Spatiotemporal Control (ProST). While previous methods such as Text2QR [34], Face2QR [5], and GladCoder [39] employ a fixed control signal throughout inference, we introduce ProST, a strategy that adaptively updates the HLG map during inference. To assess its effectiveness, we compare ProST against static control baselines using fixed activation vectors: $\beta_k = b_2 = (0, 0, 1)$ for light control and $\beta_k = b_8 = (1, 1, 1)$ for strong control. As shown in Table 9, static light control yields high visual fidelity but poor scannability, whereas strong control improves scannability at the cost of degraded aesthetics. In contrast, our ProST strategy with adaptive β_k achieves a superior trade-off, delivering outputs with both robust scannability and high visual quality.

5 Conclusion

We present AnimateQR, a generative framework for animated QR code synthesis. Leveraging hierarchical luminance guidance and progressive spatiotemporal control, AnimateQR achieves a strong balance between visual aesthetics and scannability. Extensive experiments confirm its superiority over prior methods. Beyond QR codes, our adaptive control strategy holds promise for other scannable media, such as animated AR markers, and inspires future work on constraint-aware generative modeling.

Limitations. Despite its strengths, AnimateQR has some limitations. First, generating high-resolution, visually complex animations can be computationally demanding. Second, although designed for broad compatibility, a small fraction of legacy or less-common scanners may still encounter decoding issues.

Broader Impact. By elevating the visual appeal and personal relevance of QR codes, our work re-imagines them as expressive, aesthetic artifacts—unlocking new applications in entertainment, social media, marketing, and personal memorabilia.

6 Acknowledgment

The work was supported by the National Natural Science Foundation of China under Grant 62301310.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12):2481–2495, 2017.

[2] Xianfu Cheng, Wei Zhang, Shiwei Zhang, Jian Yang, Xiangyuan Guan, Xianjie Wu, Xiang Li, Ge Zhang, Jiaheng Liu, Yuying Mai, et al. Simplevqa: Multimodal factuality evaluation for multimodal large language models. arXiv preprint arXiv:2502.13059, 2025.

[3] Hung-Kuo Chu, Chia-Sheng Chang, Ruen-Rone Lee, and Niloy J Mitra. Halftone QR Codes. ACM Transactions on Graphics (TOG), 32(6):1–8, 2013.

[4] Russ Cox. Qartcodes. https://research.swtch.com/qart, 2012.

[5] Xuehao Cui, Guangyang Wu, Zhenghao Gan, Guangtao Zhai, and Xiaohong Liu. Face2qr: A unified framework for aesthetic, face-preserving, and scannable qr code generation. arXiv preprint arXiv:2411.19246, 2024.

[6] Anthony Fu. Stylistic qr code with stable diffusion. https://antfu.me/posts/ai-qrcode, 2023.

[7] Gonzalo J Garateguy, Gonzalo R Arce, Daniel L Lau, and Ofelia P Villarreal. QR Images: Optimized Image Embedding in QR Codes. IEEE Transactions on Image Processing, 23(7):2842–2853, 2014.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. Communications of the ACM, pages 139–144, 2020.

[9] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. arXiv preprint arXiv:2307.04725, 2023.

[10] Yuan Huang, Peng Cao, and Guangwu Lyu. A directly readable halftone multifunctional color qr code. Chinese Journal of Electronics, 32(3):474–484, 2023.

[11] Yipo Huang, Xiangfei Sheng, Zhichao Yang, Quan Yuan, Zhichao Duan, Pengfei Chen, Leida Li, Weisi Lin, and Guangming Shi. Aesexpert: Towards multi-modality foundation model for image aesthetics perception. In Proceedings of the 32nd ACM International Conference on Multimedia, pages 5911–5920, 2024.

[12] Yipo Huang, Quan Yuan, Xiangfei Sheng, Zhichao Yang, Haoning Wu, Pengfei Chen, Yuzhe Yang, Leida Li, and Weisi Lin. Aesbench: An expert benchmark for multimodal large language models on image aesthetics perception. arXiv preprint arXiv:2401.08276, 2024.

[13] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014.

[14] Li Li, Jinxia Qiu, Jianfeng Lu, and Chin-Chen Chang. An aesthetic qr code solution based on error correction mechanism. Journal of Systems and Software, 116:85–94, 2016.

[15] Wenhao Li, Guangyang Wu, Wenyi Wang, Peiran Ren, and Xiaohong Liu. Fastllve: Real-time low-light video enhancement with intensity-aware look-up table. In ACM Int. Conf. Multimedia, 2023.

[16] Shih-Syun Lin, Min-Chun Hu, Chien-Han Lee, and Tong-Yee Lee. Efficient qr code beautification with high quality visual content. IEEE Transactions on Multimedia, 17(9):1515–1524, 2015.

[17] Xiaohong Liu, Lingshi Kong, Yang Zhou, Jiying Zhao, and Jun Chen. End-to-end trainable video super-resolution based on a new mechanism for implicit motion estimation and compensation. In Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), 2020.

[18] Xiaohong Liu, Yongrui Ma, Zhihao Shi, and Jun Chen. Griddehazenet: Attention-based multi-scale network for image dehazing. In Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2019.

[19] Xiaohong Liu, Kangdi Shi, Zhe Wang, and Jun Chen. Exploit camera raw data for video super- resolution via hidden markov model inference. IEEE Trans. Image Process., 30:2127–2140, 2021.

[20] Xiaohong Liu, Zhihao Shi, Zijun Wu, Jun Chen, and Guangtao Zhai. Griddehazenet+: An enhanced multi-scale network with intra-task knowledge transfer for single image dehazing. IEEE Trans. Intell. Transp. Syst., 24(1):870–884, 2023.

[21] Maxim Maximov, Ismail Elezi, and Laura Leal-Taixé. Ciagan: Conditional identity anonymization generative adversarial networks. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020.

[22] Hao Ni, Boyu Chen, Zhaohan Wang, and Zhiyong Chen. QRBTF. http://qrbtf.com, 2023.

[23] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. arXiv preprint arXiv:2112.10741, 2021.

[24] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. arXiv preprint arXiv:2204.06125, 1(2):3, 2022.

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2022.

[26] Zhihao Shi, Xiaohong Liu, Chengqi Li, Linhui Dai, Jun Chen, Timothy N. Davidson, and Jiying Zhao. Learning for unconstrained space-time video super-resolution. IEEE Trans. Broadcast., 68(2):345–358, 2022.

[27] Zhihao Shi, Xiaohong Liu, Kangdi Shi, Linhui Dai, and Jun Chen. Video frame interpolation via generalized deformable convolution. IEEE Trans. Multim., 24:426–439, 2022.

[28] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. Video frame interpolation transformer. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2022.

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502, 2020.

[30] Hao Su, Jianwei Niu, Xuefeng Liu, Qingfeng Li, Ji Wan, and Mingliang Xu. Q-Art Code: Generating Scanning-robust Art-style QR Codes by Deformable Convolution. In ACM Int. Conf. Multimedia, 2021.

[31] Hao Su, Jianwei Niu, Xuefeng Liu, Qingfeng Li, Ji Wan, Mingliang Xu, and Tao Ren. Artcoder: an end-to-end method for generating scanning-robust stylized qr codes. In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2021.

[32] Wenyi Wang, Guangyang Wu, Weitong Cai, Liaoyuan Zeng, and Jianwen Chen. Robust prior-based single image super resolution under multiple gaussian degradations. IEEE Access, 8:74195–74204, 2020.

[33] Shaoguo Wen and Junle Wang. A strong baseline for image and video quality assessment. arXiv preprint arXiv:2111.07104, 2021.

[34] Guangyang Wu, Xiaohong Liu, Jun Jia, Xuehao Cui, and Guangtao Zhai. Text2qr: Harmonizing aesthetic customization and scanning robustness for text-guided qr code generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8456–8465, 2024.

[35] Guangyang Wu, Xiaohong Liu, Kunming Luo, Xi Liu, Qingqing Zheng, Shuaicheng Liu, Xinyang Jiang, Guangtao Zhai, and Wenyi Wang. Accflow: Backward accumulation for long-range optical flow. In Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2023.

[36] Guangyang Wu, Lili Zhao, Wenyi Wang, Liaoyuan Zeng, and Jianwen Chen. Pred: A parallel network for handling multiple degradations via single model in single image super-resolution. In Proc. IEEE Int. Conf. Image Process. (ICIP), 2019.

[37] Haoning Wu, Zicheng Zhang, Erli Zhang, Chaofeng Chen, Liang Liao, Annan Wang, Chunyi Li, Wenxiu Sun, Qiong Yan, Guangtao Zhai, et al. Q-bench: A benchmark for general-purpose foundation models on low-level vision. arXiv preprint arXiv:2309.14181, 2023.

[38] Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, et al. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. arXiv preprint arXiv:2312.17090, 2023.

[39] Yuqiu Xie, Bolin Jiang, Jiawei Li, Naiqi Li, Bin Chen, Tao Dai, Yuang Peng, and Shu-Tao Xia. Gladcoder: stylized qr code generation with grayscale-aware denoising process. In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, pages 7780–7787, 2024.

[40] Mingliang Xu, Qingfeng Li, Jianwei Niu, Hao Su, Xiting Liu, Weiwei Xu, Pei Lv, Bing Zhou, and Yi Yang. ART-UP: A novel method for generating scanning-robust aesthetic QR codes. ACM Trans. Multim. Comput. Commun. Appl., 17(1):25:1–25:23, 2021.

[41] Mingliang Xu, Hao Su, Yafei Li, Xi Li, Jing Liao, Jianwei Niu, Pei Lv, and Bing Zhou. Stylized aesthetic qr code. IEEE Transactions on Multimedia, 21(8):1960–1970, 2019.

[42] Liming Zhai, Qing Guo, Xiaofei Xie, Lei Ma, Yi Estelle Wang, and Yang Liu. A3gan: Attribute-aware anonymization networks for face de-identification. In Proceedings of the 30th ACM International Conference on Multimedia, pages 5303–5313, 2022.

[43] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In Proceedings of the IEEE/CVF international conference on computer vision, pages 3836–3847, 2023.

[44] Weixia Zhang, Guangtao Zhai, Ying Wei, Xiaokang Yang, and Kede Ma. Blind image quality assessment via vision-language correspondence: A multitask learning perspective. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 14071–14081, 2023.

[45] Yongtai Zhang, Shihong Deng, Zhihong Liu, and Yongtao Wang. Aesthetic qr codes based on two-stage image blending. In MultiMedia Modeling: 21st International Conference, MMM 2015, Sydney, NSW, Australia, January 5-7, 2015, Proceedings, Part II 21, pages 183–194. Springer, 2015.

[46] Sizhe Zheng, Pan Gao, Peng Zhou, and Jie Qin. Puff-net: Efficient style transfer with pure content and style feature fusion network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8059–8068, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In abstract, the main contributions of this paper are emphasized. Furthermore, in the last paragraph of the introduction, these contributions are clearly listed again.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Section 5.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all needed information to reproduce the main experimental results of this paper in Section 4. Our code will be released upon publication.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Although no link to the code is currently provided in the paper, we have a strong intention to follow up by releasing the code of this work via Github to promote the development of related areas once the paper is published.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All experiments details are illustrated in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: This paper mainly conducts qualitative comparisons and subjective experiments. Therefore, the corresponding error bars are not applicable.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computational resources have been described in Section 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: This work is conducted in accordance with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to the Section 5.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, are properly credited, and the license and terms of use are explicitly mentioned and are properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets introduced in the paper are well documented alongside the assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: This paper includes the full text of instructions given to participants and screenshots, and the human subjects are paid at least the minimum wage in the country of the data collector, following the NeurIPS Code of Ethics.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: There is no such potential risks aware for research with human subjects in this paper. We have obtained the IRB approval and also adhere to the NeurIPS Code of Ethics.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLM is used only for writing, editing, and formatting purposes, without impact the core methodology, scientific rigorousness, or originality of this research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.