README

SBER-MoVQGAN

Framework PyTorch | Huggingface | Open in Colab

Habr post

https://github.com/ai-forever/MoVQGAN

MoVQ: Modulating Quantized Vectors for High-Fidelity Image Generation, arXiv:2209.09002

MoVQ: Modulating Quantized Vectors for High-Fidelity Image Generation

Chuanxia Zheng
Monash University
chuanxiazheng@gmail.com

Long Tung Vuong
VinAI
longvt94@gmail.com

Jianfei Cai
Monash University
Jianfei.Cai@monash.edu

Dinh Phung
Monash University
dinh.phung@monash.edu

Background. Given an image $x \in \mathbb{R}^{H \times W \times 3}$, vanilla VQ-VAEs learn a discrete codebook to represent observations as a collection of codebook entries $z_q \in \mathbb{R}^{h \times w \times n_q}$, where n_q is the dimensionality of quantized vectors in the codebook. In this way, each image can be equivalently represented as a compact sequence $\mathbf{s}$ with $h \cdot w$ indices of the codevectors z_q. Formally, the observed image x is reconstructed by:

$$\hat{x} = \mathcal{G}_\theta(z_q) = \mathcal{G}_\theta(\mathbf{q}(\hat{z})) = \mathcal{G}_\theta(\mathbf{q}(\mathcal{E}_\psi(x))). \tag{1}$$

In particular, an encoder $\mathcal{E}_\psi(\cdot)$ first embeds an image x into a continuous vector $\hat{z}$, and the quantization operator $\mathbf{q}(\cdot)$ is then conducted to transfer the continuous feature $\hat{z}$ into the discrete space by looking up the closest codebook entry z_k for each spatial grid feature $\hat{z}_{ij}$ within $\hat{z}$:

$$z_q = \mathbf{q}(\hat{z}) = \arg\min_{z_k \in \mathcal{Z}} \|\hat{z}_{ij} - z_k\|, \tag{2}$$

where $\mathcal{Z} \in \mathbb{R}^{K \times n_q}$ is the codebook that consists of K entries with n_q dimensions. The quantized vector z_q is finally transmitted to a decoder $\mathcal{G}_\theta(\cdot)$ for rebuilding the original image. The overall models and the codebook can be learned by optimizing the following objective:

$$\mathcal{L}(\mathcal{E}_\psi, \mathcal{G}_\theta, \mathcal{Z}) = \|x - \hat{x}\|_2^2 + \|\mathrm{sg}[\mathcal{E}_\psi(x)] - z_q\|_2^2 + \beta\|\mathrm{sg}[z_q] - \mathcal{E}_\psi(x)\|_2^2. \tag{3}$$

Here, sg denotes the stop-gradient operator, and β is a hyperparameter for the third *commitment loss*. The first term is a *reconstruction loss* to estimate the error between the observed x and the reconstructed $\hat{x}$. The second term is the *codebook loss* to optimize the entries in the codebook. We use the released VQGAN implementation as our baseline.

Spatially conditional normalization. The quantization operator is lossy [31], and similar patches

The structure of our modulated decoder is illustrated in Fig. 2 (left), with an activation F normalized by a conventional normalization, and then modulated by the learned scale and bias calculated from the embedded vector. Specifically, the activation F of the i-th layer in the decoder $\mathcal{G}_\theta$ is given by:

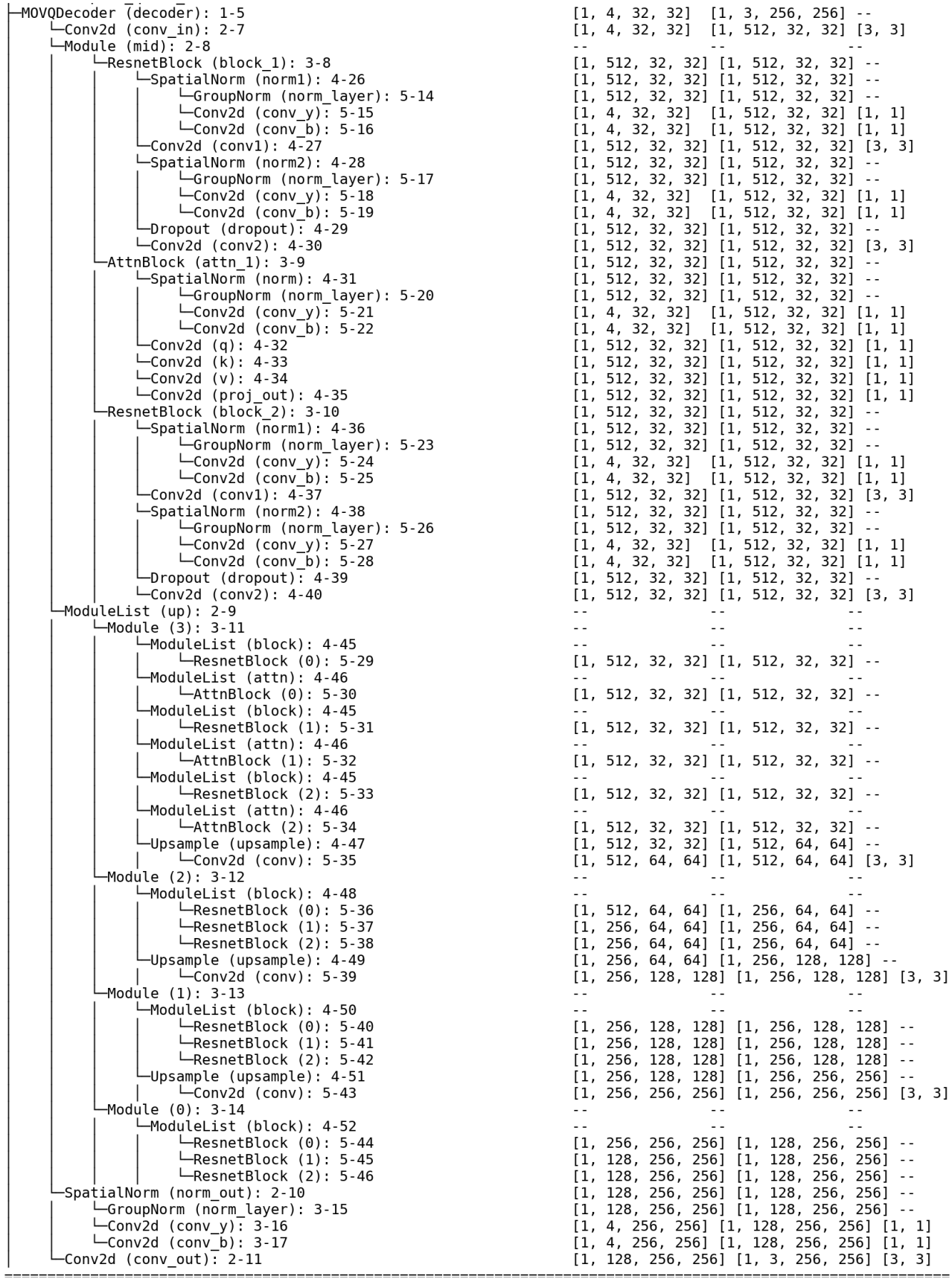
$$F^i = \phi_\gamma(z_q)\frac{F^{i-1} - \mu(F^{i-1})}{\sigma(F^{i-1})} + \phi_\beta(z_q), \tag{4}$$

Model structure:

```
==========================================================================================================================
Layer (type (var_name):depth-idx)                                       Input Shape     Output Shape    Kernel Shape
==========================================================================================================================
MOVQ (MOVQ)                                                             [1, 3, 256, 256] [1, 3, 256, 256] --
├─Encoder (encoder): 1-1                                                [1, 3, 256, 256] [1, 4, 32, 32]  --
├─Conv2d (quant_conv): 1-2                                              [1, 4, 32, 32]  [1, 4, 32, 32]  [1, 1]
├─VectorQuantizer (quantize): 1-3                                       [1, 4, 32, 32]  [1, 4, 32, 32]  --
├─Conv2d (post_quant_conv): 1-4                                         [1, 4, 32, 32]  [1, 4, 32, 32]  [1, 1]
├─MOVQDecoder (decoder): 1-5                                            [1, 4, 32, 32]  [1, 3, 256, 256] --
==========================================================================================================================
Total params: 67,832,495
Trainable params: 67,832,495
Non-trainable params: 0
Total mult-adds (Units.GIGABYTES): 343.09
==========================================================================================================================
Input size (MB): 0.79
Forward/backward pass size (MB): 4716.10
Params size (MB): 271.33
Estimated Total Size (MB): 4988.22
==========================================================================================================================
```

```
==========================================================================================================================
Layer (type (var_name):depth-idx)                                       Input Shape     Output Shape    Kernel Shape
==========================================================================================================================
MOVQ (MOVQ)                                                             [1, 3, 256, 256] [1, 3, 256, 256] --
├─Encoder (encoder): 1-1                                                [1, 3, 256, 256] [1, 4, 32, 32]  --
│    └─Conv2d (conv_in): 2-1                                            [1, 3, 256, 256] [1, 128, 256, 256] [3, 3]
│    └─ModuleList (down): 2-2                                           --              --              --
│    │    └─Module (0): 3-1                                             --              --              --
│    │    │    └─ModuleList (block): 4-1                                --              --              --
│    │    │    │    └─ResnetBlock (0): 5-1                              [1, 128, 256, 256] [1, 128, 256, 256] --
│    │    │    │    └─ResnetBlock (1): 5-2                              [1, 128, 256, 256] [1, 128, 256, 256] --
│    │    │    └─Downsample (downsample): 4-2                           [1, 128, 256, 256] [1, 128, 128, 128] --
│    │    │    │    └─Conv2d (conv): 5-3                                [1, 128, 257, 257] [1, 128, 128, 128] [3, 3]
│    │    └─Module (1): 3-2                                             --              --              --
│    │    │    └─ModuleList (block): 4-3                                --              --              --
│    │    │    │    └─ResnetBlock (0): 5-4                              [1, 128, 128, 128] [1, 256, 128, 128] --
│    │    │    │    └─ResnetBlock (1): 5-5                              [1, 256, 128, 128] [1, 256, 128, 128] --
│    │    │    └─Downsample (downsample): 4-4                           [1, 256, 128, 128] [1, 256, 64, 64] --
│    │    │    │    └─Conv2d (conv): 5-6                                [1, 256, 129, 129] [1, 256, 64, 64] [3, 3]
│    │    └─Module (2): 3-3                                             --              --              --
│    │    │    └─ModuleList (block): 4-5                                --              --              --
│    │    │    │    └─ResnetBlock (0): 5-7                              [1, 256, 64, 64] [1, 256, 64, 64] --
│    │    │    │    └─ResnetBlock (1): 5-8                              [1, 256, 64, 64] [1, 256, 64, 64] --
│    │    │    └─Downsample (downsample): 4-6                           [1, 256, 64, 64] [1, 256, 32, 32] --
│    │    │    │    └─Conv2d (conv): 5-9                                [1, 256, 65, 65] [1, 256, 32, 32] [3, 3]
│    │    └─Module (3): 3-4                                             --              --              --
│    │    │    └─ModuleList (block): 4-9                                --              --              --
│    │    │    │    └─ResnetBlock (0): 5-10                             [1, 256, 32, 32] [1, 512, 32, 32] --
│    │    │    └─ModuleList (attn): 4-10                                --              --              --
│    │    │    │    └─AttnBlock (0): 5-11                               [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─ModuleList (block): 4-9                                --              --              --
│    │    │    │    └─ResnetBlock (1): 5-12                             [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─ModuleList (attn): 4-10                                --              --              --
│    │    │    │    └─AttnBlock (1): 5-13                               [1, 512, 32, 32] [1, 512, 32, 32] --
│    └─Module (mid): 2-3                                                --              --              --
│    │    └─ResnetBlock (block_1): 3-5                                  [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─GroupNorm (norm1): 4-11                                [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Conv2d (conv1): 4-12                                   [1, 512, 32, 32] [1, 512, 32, 32] [3, 3]
│    │    │    └─GroupNorm (norm2): 4-13                                [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Dropout (dropout): 4-14                                [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Conv2d (conv2): 4-15                                   [1, 512, 32, 32] [1, 512, 32, 32] [3, 3]
│    │    └─AttnBlock (attn_1): 3-6                                     [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─GroupNorm (norm): 4-16                                 [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Conv2d (q): 4-17                                       [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
│    │    │    └─Conv2d (k): 4-18                                       [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
│    │    │    └─Conv2d (v): 4-19                                       [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
│    │    │    └─Conv2d (proj_out): 4-20                                [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
│    │    └─ResnetBlock (block_2): 3-7                                  [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─GroupNorm (norm1): 4-21                                [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Conv2d (conv1): 4-22                                   [1, 512, 32, 32] [1, 512, 32, 32] [3, 3]
│    │    │    └─GroupNorm (norm2): 4-23                                [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Dropout (dropout): 4-24                                [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Conv2d (conv2): 4-25                                   [1, 512, 32, 32] [1, 512, 32, 32] [3, 3]
│    └─GroupNorm (norm_out): 2-4                                        [1, 512, 32, 32] [1, 512, 32, 32] --
│    └─Conv2d (conv_out): 2-5                                           [1, 512, 32, 32] [1, 4, 32, 32]  [3, 3]
```

```
├─Conv2d (quant_conv): 1-2                                [1, 4, 32, 32]  [1, 4, 32, 32]  [1, 1]
├─VectorQuantizer (quantize): 1-3                         [1, 4, 32, 32]  [1, 4, 32, 32]  --
│    └─Embedding (embedding): 2-6                         [1024]          [1024, 4]       --
├─Conv2d (post quant conv): 1-4                           [1, 4, 32, 32]  [1, 4, 32, 32]  [1, 1]
```


├─MOVQDecoder (decoder): 1-5 [1, 4, 32, 32] [1, 3, 256, 256] --
└─Conv2d (conv_in): 2-7 [1, 4, 32, 32] [1, 512, 32, 32] [3, 3]
└─Module (mid): 2-8 -- -- --
└─ResnetBlock (block_1): 3-8 [1, 512, 32, 32] [1, 512, 32, 32] --
└─SpatialNorm (norm1): 4-26 [1, 512, 32, 32] [1, 512, 32, 32] --
└─GroupNorm (norm_layer): 5-14 [1, 512, 32, 32] [1, 512, 32, 32] --
└─Conv2d (conv_y): 5-15 [1, 4, 32, 32] [1, 512, 32, 32] [1, 1]
└─Conv2d (conv_b): 5-16 [1, 4, 32, 32] [1, 512, 32, 32] [1, 1]
└─Conv2d (conv1): 4-27 [1, 512, 32, 32] [1, 512, 32, 32] [3, 3]
└─SpatialNorm (norm2): 4-28 [1, 512, 32, 32] [1, 512, 32, 32] --
└─GroupNorm (norm_layer): 5-17 [1, 512, 32, 32] [1, 512, 32, 32] --
└─Conv2d (conv_y): 5-18 [1, 4, 32, 32] [1, 512, 32, 32] [1, 1]
└─Conv2d (conv_b): 5-19 [1, 4, 32, 32] [1, 512, 32, 32] [1, 1]
└─Dropout (dropout): 4-29 [1, 512, 32, 32] [1, 512, 32, 32] --
└─Conv2d (conv2): 4-30 [1, 512, 32, 32] [1, 512, 32, 32] [3, 3]
└─AttnBlock (attn_1): 3-9 [1, 512, 32, 32] [1, 512, 32, 32] --
└─SpatialNorm (norm): 4-31 [1, 512, 32, 32] [1, 512, 32, 32] --
└─GroupNorm (norm_layer): 5-20 [1, 512, 32, 32] [1, 512, 32, 32] --
└─Conv2d (conv_y): 5-21 [1, 4, 32, 32] [1, 512, 32, 32] [1, 1]
└─Conv2d (conv_b): 5-22 [1, 4, 32, 32] [1, 512, 32, 32] [1, 1]
└─Conv2d (q): 4-32 [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
└─Conv2d (k): 4-33 [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
└─Conv2d (v): 4-34 [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
└─Conv2d (proj_out): 4-35 [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
└─ResnetBlock (block_2): 3-10 [1, 512, 32, 32] [1, 512, 32, 32] --
└─SpatialNorm (norm1): 4-36 [1, 512, 32, 32] [1, 512, 32, 32] --
└─GroupNorm (norm_layer): 5-23 [1, 512, 32, 32] [1, 512, 32, 32] --
└─Conv2d (conv_y): 5-24 [1, 4, 32, 32] [1, 512, 32, 32] [1, 1]
└─Conv2d (conv_b): 5-25 [1, 4, 32, 32] [1, 512, 32, 32] [1, 1]
└─Conv2d (conv1): 4-37 [1, 512, 32, 32] [1, 512, 32, 32] [3, 3]
└─SpatialNorm (norm2): 4-38 [1, 512, 32, 32] [1, 512, 32, 32] --
└─GroupNorm (norm_layer): 5-26 [1, 512, 32, 32] [1, 512, 32, 32] --
└─Conv2d (conv_y): 5-27 [1, 4, 32, 32] [1, 512, 32, 32] [1, 1]
└─Conv2d (conv_b): 5-28 [1, 4, 32, 32] [1, 512, 32, 32] [1, 1]
└─Dropout (dropout): 4-39 [1, 512, 32, 32] [1, 512, 32, 32] --
└─Conv2d (conv2): 4-40 [1, 512, 32, 32] [1, 512, 32, 32] [3, 3]
└─ModuleList (up): 2-9 -- -- --
└─Module (3): 3-11 -- -- --
└─ModuleList (block): 4-45 -- -- --
└─ResnetBlock (0): 5-29 [1, 512, 32, 32] [1, 512, 32, 32] --
└─ModuleList (attn): 4-46 -- -- --
└─AttnBlock (0): 5-30 [1, 512, 32, 32] [1, 512, 32, 32] --
└─ModuleList (block): 4-45 -- -- --
└─ResnetBlock (1): 5-31 [1, 512, 32, 32] [1, 512, 32, 32] --
└─ModuleList (attn): 4-46 -- -- --
└─AttnBlock (1): 5-32 [1, 512, 32, 32] [1, 512, 32, 32] --
└─ModuleList (block): 4-45 -- -- --
└─ResnetBlock (2): 5-33 [1, 512, 32, 32] [1, 512, 32, 32] --
└─ModuleList (attn): 4-46 -- -- --
└─AttnBlock (2): 5-34 [1, 512, 32, 32] [1, 512, 32, 32] --
└─Upsample (upsample): 4-47 [1, 512, 32, 32] [1, 512, 64, 64] --
└─Conv2d (conv): 5-35 [1, 512, 64, 64] [1, 512, 64, 64] [3, 3]
└─Module (2): 3-12 -- -- --
└─ModuleList (block): 4-48 -- -- --
└─ResnetBlock (0): 5-36 [1, 512, 64, 64] [1, 256, 64, 64] --
└─ResnetBlock (1): 5-37 [1, 256, 64, 64] [1, 256, 64, 64] --
└─ResnetBlock (2): 5-38 [1, 256, 64, 64] [1, 256, 64, 64] --
└─Upsample (upsample): 4-49 [1, 256, 64, 64] [1, 256, 128, 128] --
└─Conv2d (conv): 5-39 [1, 256, 128, 128] [1, 256, 128, 128] [3, 3]
└─Module (1): 3-13 -- -- --
└─ModuleList (block): 4-50 -- -- --
└─ResnetBlock (0): 5-40 [1, 256, 128, 128] [1, 256, 128, 128] --
└─ResnetBlock (1): 5-41 [1, 256, 128, 128] [1, 256, 128, 128] --
└─ResnetBlock (2): 5-42 [1, 256, 128, 128] [1, 256, 128, 128] --
└─Upsample (upsample): 4-51 [1, 256, 128, 128] [1, 256, 256, 256] --
└─Conv2d (conv): 5-43 [1, 256, 256, 256] [1, 256, 256, 256] [3, 3]
└─Module (0): 3-14 -- -- --
└─ModuleList (block): 4-52 -- -- --
└─ResnetBlock (0): 5-44 [1, 256, 256, 256] [1, 128, 256, 256] --
└─ResnetBlock (1): 5-45 [1, 128, 256, 256] [1, 128, 256, 256] --
└─ResnetBlock (2): 5-46 [1, 128, 256, 256] [1, 128, 256, 256] --
└─SpatialNorm (norm_out): 2-10 [1, 128, 256, 256] [1, 128, 256, 256] --
└─GroupNorm (norm_layer): 3-15 [1, 128, 256, 256] [1, 128, 256, 256] --
└─Conv2d (conv_y): 3-16 [1, 4, 256, 256] [1, 128, 256, 256] [1, 1]
└─Conv2d (conv_b): 3-17 [1, 4, 256, 256] [1, 128, 256, 256] [1, 1]
└─Conv2d (conv_out): 2-11 [1, 128, 256, 256] [1, 3, 256, 256] [3, 3]

```
│    │    └─ResnetBlock (block_1): 3-5                  [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─GroupNorm (norm1): 4-11                [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Conv2d (conv1): 4-12                   [1, 512, 32, 32] [1, 512, 32, 32] [3, 3]
│    │    │    └─GroupNorm (norm2): 4-13                [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Dropout (dropout): 4-14                [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Conv2d (conv2): 4-15                   [1, 512, 32, 32] [1, 512, 32, 32] [3, 3]
│    │    └─AttnBlock (attn_1): 3-6                     [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─GroupNorm (norm): 4-16                 [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Conv2d (q): 4-17                       [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
│    │    │    └─Conv2d (k): 4-18                       [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
│    │    │    └─Conv2d (v): 4-19                       [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
│    │    │    └─Conv2d (proj_out): 4-20                [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]

│    │    └─ResnetBlock (block_1): 3-8                  [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─SpatialNorm (norm1): 4-26              [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Conv2d (conv1): 4-27                   [1, 512, 32, 32] [1, 512, 32, 32] [3, 3]
│    │    │    └─SpatialNorm (norm2): 4-28              [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Dropout (dropout): 4-29                [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Conv2d (conv2): 4-30                   [1, 512, 32, 32] [1, 512, 32, 32] [3, 3]
│    │    └─AttnBlock (attn_1): 3-9                     [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─SpatialNorm (norm): 4-31               [1, 512, 32, 32] [1, 512, 32, 32] --
│    │    │    └─Conv2d (q): 4-32                       [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
│    │    │    └─Conv2d (k): 4-33                       [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
│    │    │    └─Conv2d (v): 4-34                       [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
│    │    │    └─Conv2d (proj_out): 4-35                [1, 512, 32, 32] [1, 512, 32, 32] [1, 1]
```