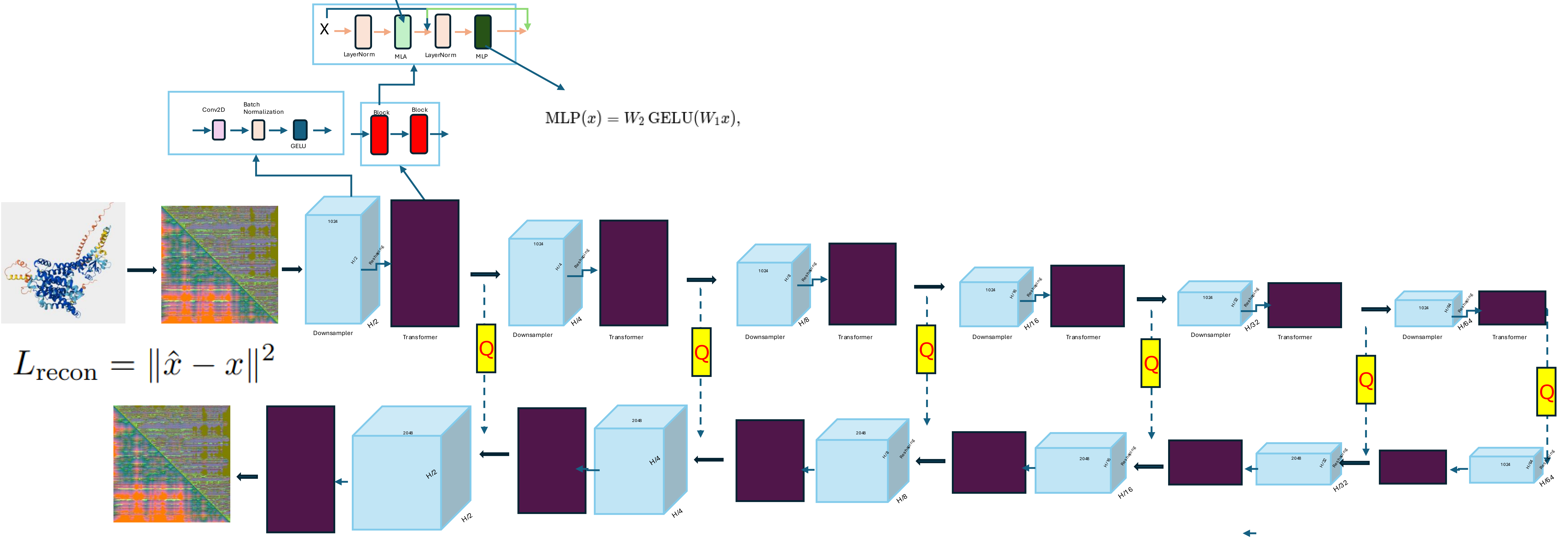
$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

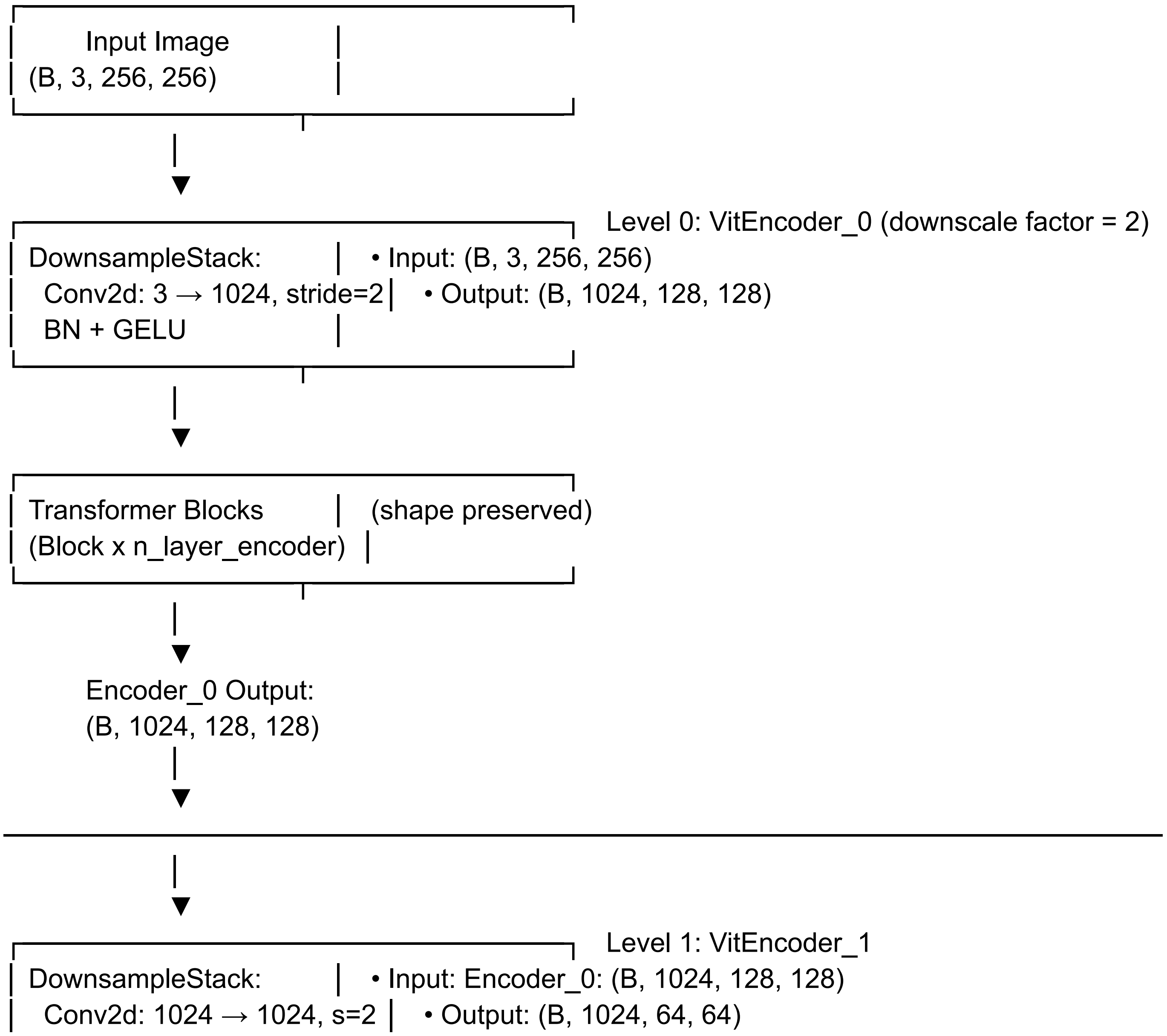
$$\mathrm{MLP}(x) = W_2\,\mathrm{GELU}(W_1 x),$$



$$L_{\text{recon}} = \|\hat{x} - x\|^2$$

$$L_{\text{vq}} = \sum_{\ell} \left\| \mathrm{sg}[q_\ell] - e_\ell \right\|^2$$

$$L = L_{\text{recon}} + \beta\, L_{\text{vq}}$$



Input Image
(B, 3, 256, 256)

Level 0: VitEncoder_0 (downscale factor = 2)

DownsampleStack:
Conv2d: 3 → 1024, stride=2
BN + GELU

- Input: (B, 3, 256, 256)
- Output: (B, 1024, 128, 128)

Transformer Blocks
(Block x n_layer_encoder)

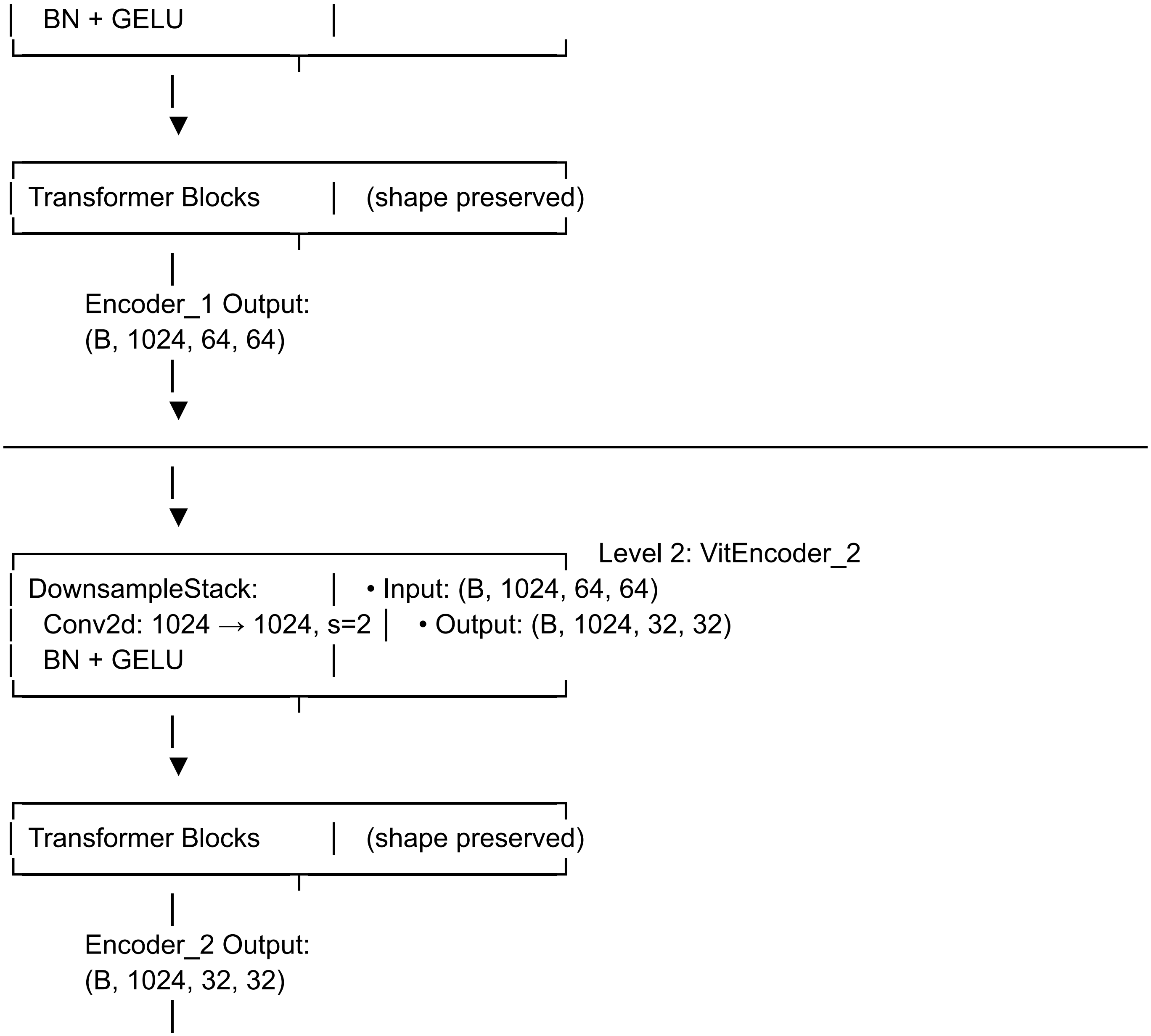
(shape preserved)

Encoder_0 Output:
(B, 1024, 128, 128)

Level 1: VitEncoder_1

DownsampleStack:
Conv2d: 1024 → 1024, s=2

- Input: Encoder_0: (B, 1024, 128, 128)
- Output: (B, 1024, 64, 64)



BN + GELU

Transformer Blocks

(shape preserved)

Encoder_1 Output:
(B, 1024, 64, 64)

Level 2: VitEncoder_2

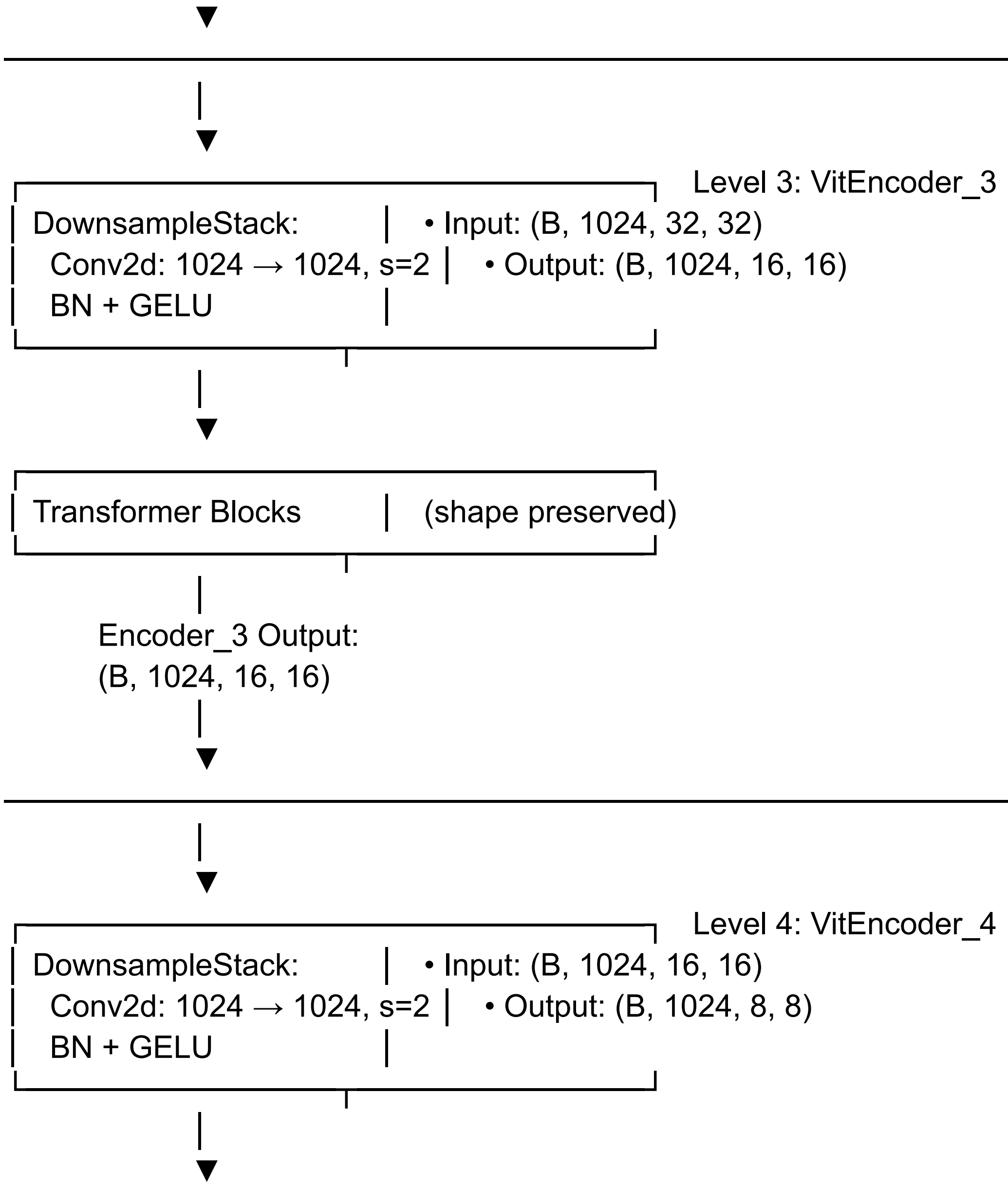
DownsampleStack:
Conv2d: 1024 → 1024, s=2
BN + GELU

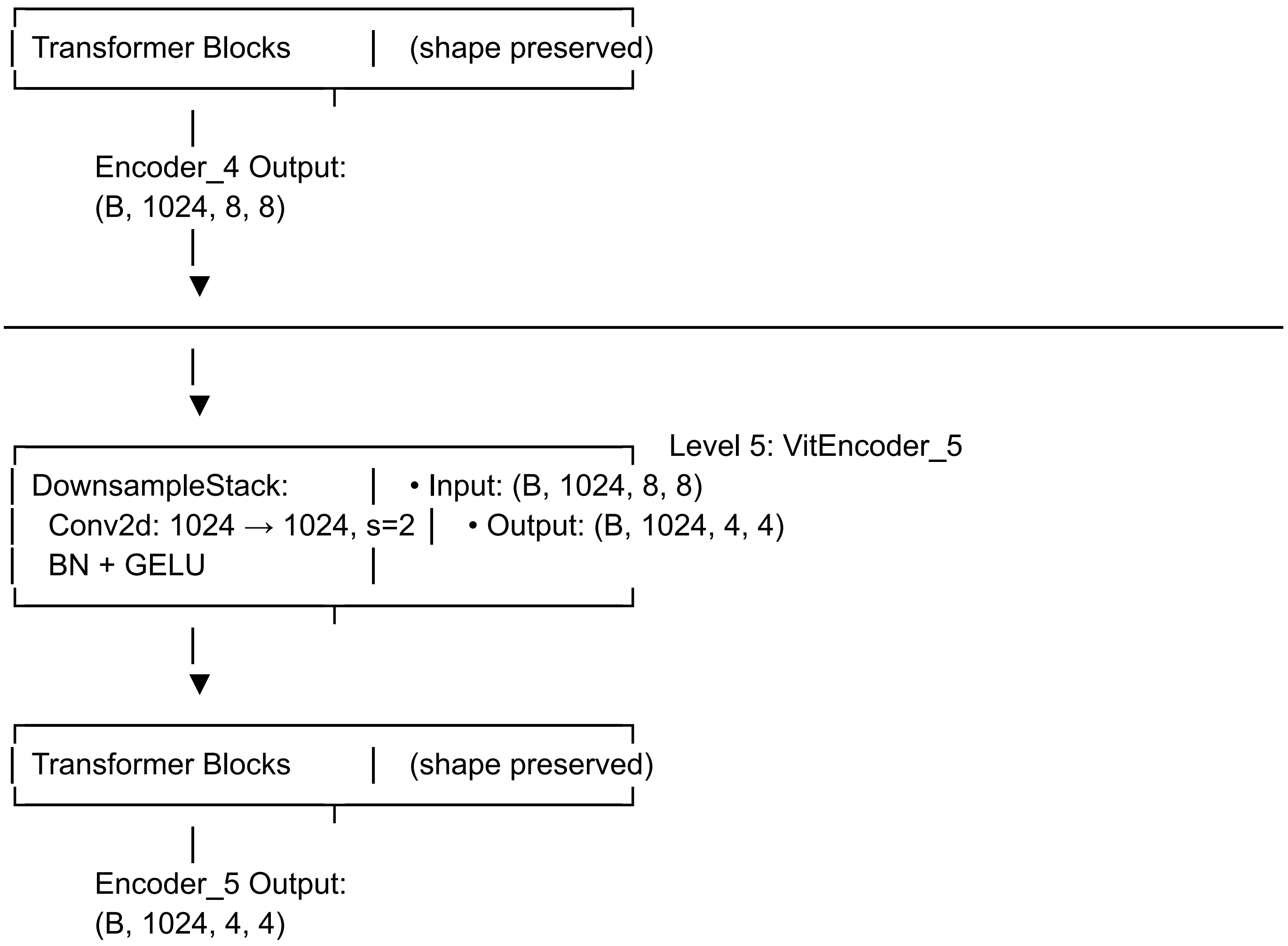
• Input: (B, 1024, 64, 64)
• Output: (B, 1024, 32, 32)

Transformer Blocks

(shape preserved)

Encoder_2 Output:
(B, 1024, 32, 32)


Level 3: VitEncoder_3
DownsampleStack:
Conv2d: 1024 → 1024, s=2
BN + GELU
• Input: (B, 1024, 32, 32)
• Output: (B, 1024, 16, 16)
Transformer Blocks
(shape preserved)
Encoder_3 Output:
(B, 1024, 16, 16)
Level 4: VitEncoder_4
DownsampleStack:
Conv2d: 1024 → 1024, s=2
BN + GELU
• Input: (B, 1024, 16, 16)
• Output: (B, 1024, 8, 8)



Level 5 (Coarsest):

```
Input: Encoder_5 from bottom-up → (B, 1024, 4, 4)
  |
  ▼
[Codebook_5]:
  • For top level, in_ch = 1024.
  • Quantization produces:
        - q_5: (B, 1024, 4, 4)
        - Code indices: (B, 4, 4)
  |
  ▼
No lower-level code yet, so:
  dec_input_5 = q_5  (shape: (B, 1024, 4, 4))
  |
  ▼
[VitDecoder_5]:
  • 1×1 Conv: (B, 1024, 4, 4) → remains (B, 1024, 4, 4)
  • Transformer Blocks: (B, 1024, 4, 4)
  • UpsampleStack with factor=2: (B, 1024, 4, 4) → (B, 1024, 8, 8)
  |
  ▼
Decoder_5 Output: (B, 1024, 8, 8)
  └── Append q_5 to code_outputs.
```

Level 4:

```
Input:
  • Encoder_4: (B, 1024, 8, 8)
  • Upsample previous decoder (Decoder_5) is already (B, 1024, 8, 8)
  → Concatenate along channel dim:
        cond = cat(Encoder_4, Upsampled Decoder_5) = (B, 2048, 8, 8)
  │
  ▼
[Codebook_4]:
  • For non-top levels, in_ch = 1024×2 = 2048.
  • Quantization produces:
        - q_4: (B, 1024, 8, 8)
        - Code indices: (B, 8, 8)
  │
  ▼
Also, upsample previously computed q_5 from (B, 1024, 4, 4) → (B, 1024, 8, 8)
  └─ Now, dec_input_4 = cat(q_4, upsampled q_5) = (B, 2048, 8, 8)
  │
  ▼
[VitDecoder_4]:
  • 1×1 Conv: projects (B, 2048, 8, 8) → (B, 1024, 8, 8)
  • Transformer Blocks: (B, 1024, 8, 8)
  • UpsampleStack with factor=2: (B, 1024, 8, 8) → (B, 1024, 16, 16)
  │
  ▼
Decoder_4 Output: (B, 1024, 16, 16)
  └─ Append q_4 to code_outputs.
```

Level 3:

Input:
  • Encoder_3: (B, 1024, 16, 16)

```
  • Upsample previous decoder output: (B, 1024, 16, 16)
  → Concatenate: cond = (B, 2048, 16, 16)
   │
   ▼
[Codebook_3]:
  • in_ch = 2048.
  • Quantization yields:
        - q_3: (B, 1024, 16, 16)
        - Code indices: (B, 16, 16)
   │
   ▼
Upsample lower codes (q_4, q_5) to current resolution (if needed) and concatenate:
  dec_input_3 = (B, 2048, 16, 16)
   │
   ▼
[VitDecoder_3]:
  • 1×1 Conv → (B, 1024, 16, 16)
  • Transformer Blocks → (B, 1024, 16, 16)
  • UpsampleStack with factor=2 → (B, 1024, 32, 32)
   │
   ▼
Decoder_3 Output: (B, 1024, 32, 32)
   └─ Append q_3 to code_outputs.
```

Level 2:

```
Input:
  • Encoder_2: (B, 1024, 32, 32)
  • Upsample previous decoder output: (B, 1024, 32, 32)
  → Concatenate: cond = (B, 2048, 32, 32)
   │
   ▼
[Codebook_2]:
  • in_ch = 2048.
  • Quantization yields:
        - q_2: (B, 1024, 32, 32)
        - Code indices: (B, 32, 32)
   │
   ▼
Upsample lower codes to current resolution; then:
  dec_input_2 = (B, 2048, 32, 32)
   │
   ▼
[VitDecoder_2]:
  • 1×1 Conv → (B, 1024, 32, 32)
  • Transformer Blocks → (B, 1024, 32, 32)
  • UpsampleStack with factor=2 → (B, 1024, 64, 64)
   │
   ▼
Decoder_2 Output: (B, 1024, 64, 64)
   └─ Append q_2 to code_outputs.
```

Level 1:

```
Input:
  • Encoder_1: (B, 1024, 64, 64)
  • Upsample previous decoder output: (B, 1024, 64, 64)
  → Concatenate: cond = (B, 2048, 64, 64)
   |
   ▼
[Codebook_1]:
  • in_ch = 2048.
  • Quantization yields:
        - q_1: (B, 1024, 64, 64)
        - Code indices: (B, 64, 64)
   |
   ▼
Combine with upsampled lower codes:
  dec_input_1 = (B, 2048, 64, 64)
   |
   ▼
[VitDecoder_1]:
  • 1×1 Conv → (B, 1024, 64, 64)
  • Transformer Blocks → (B, 1024, 64, 64)
  • UpsampleStack with factor=2 → (B, 1024, 128, 128)
   |
   ▼
Decoder_1 Output: (B, 1024, 128, 128)
   └─ Append q_1 to code_outputs.
```

Level 0 (Finest):

```
Input:
  • Encoder_0: (B, 1024, 128, 128)
  • Upsample previous decoder output: (B, 1024, 128, 128)
  → Concatenate: cond = (B, 2048, 128, 128)
   │
   ▼
[Codebook_0]:
  • in_ch = 2048.
  • Quantization yields:
        - q_0: (B, 1024, 128, 128)
        - Code indices: (B, 128, 128)
   │
   ▼
Combine with upsampled lower codes:
  dec_input_0 = (B, 2048, 128, 128)
   │
   ▼
[VitDecoder_0]:
  • 1×1 Conv → (B, 1024, 128, 128)
  • Transformer Blocks → (B, 1024, 128, 128)
  • UpsampleStack with factor=2 → (B, out_ch, 256, 256)
        Note: For level 0, out_ch is set to 3.
   │
   ▼
Decoder_0 Output (Reconstruction): (B, 3, 256, 256)
```
