GETRegionDiffusion Model Architecture

A Diffusion Transformer for Genomic Region-Based Masked Motif Prediction

Architecture Documentation

December 1, 2025

1 Overview

`GETRegionDiffusion` is a **Diffusion Transformer (DiT)** adapted for genomic region-based masked motif prediction. It combines the structure of a masked autoencoder with diffusion-style timestep conditioning. The model learns to reconstruct masked genomic motif patterns, conditioned on the diffusion timestep which acts as a noise level indicator.

2 Architecture Diagram

3 Key Components

3.1 Adaptive Layer Normalization (adaLN)

The core innovation from DiT. Instead of standard LayerNorm, the model uses a modulation function:

$$\text{modulate}(x, \text{shift}, \text{scale}) = x \cdot (1 + \text{scale}) + \text{shift}$$

where `shift` and `scale` are predicted from the timestep embedding. This allows the model to be conditioned on the diffusion timestep.

3.2 Diffusion Schedule

Linear noise schedule with the following parameters:

```
diffusion:
  num_timesteps: 1000
  beta_start: 0.0001
  beta_end: 0.02
```

During training, a random timestep $t \in [0, 1000)$ is sampled and used to condition the transformer.

3.3 Masked Prediction Task

- **Input:** 900 genomic regions, each with 283 motif features
- **Mask ratio:** 50% of regions are masked
- **Goal:** Predict the original motif features for masked regions

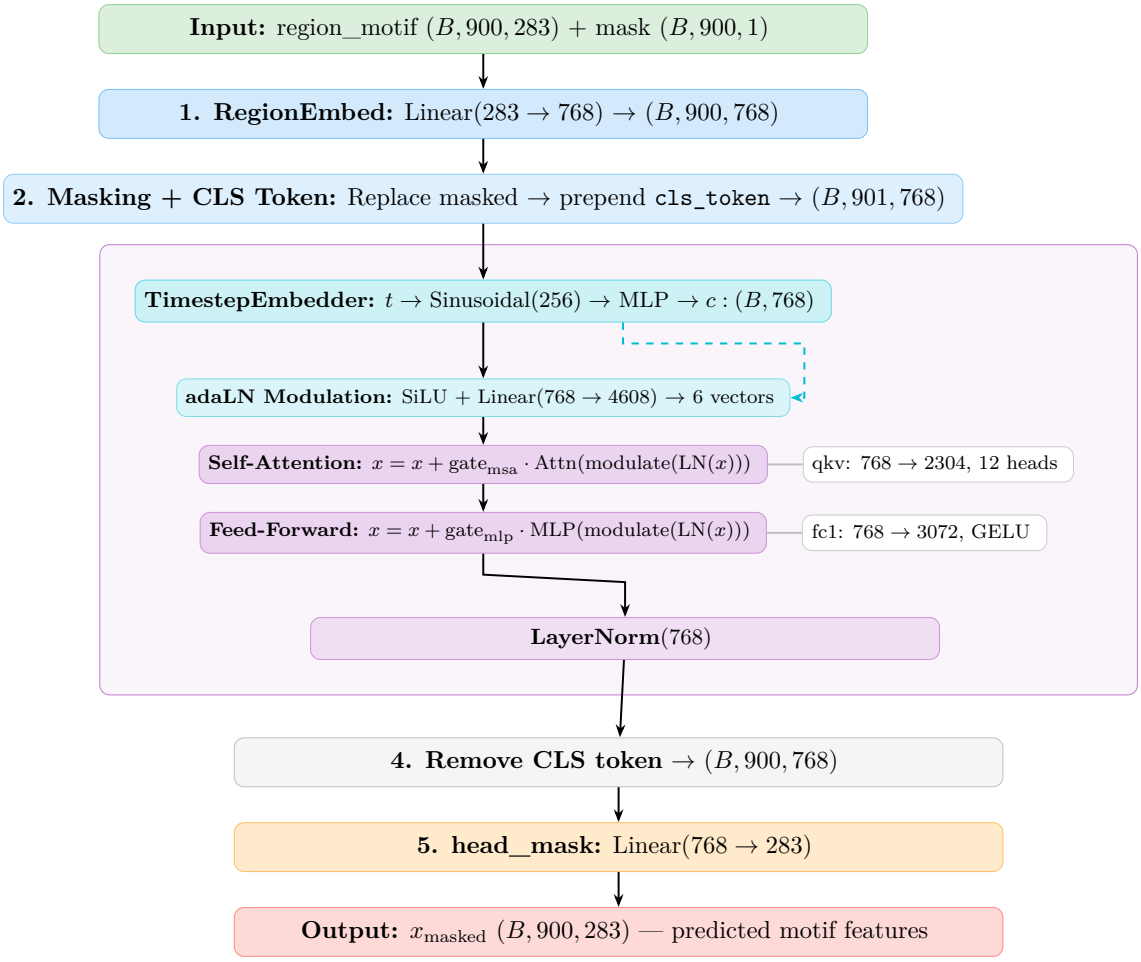

Figure 1: GETRegionDiffusion architecture. The model embeds genomic region motif features, applies masking, and processes through a DiT-style transformer with timestep conditioning via adaptive layer normalization (adaLN). The output head predicts motif features for masked regions.

Table 1: Model configuration from `dit.yaml`

Parameter	Value	Description
`num_regions`	900	Genomic regions per sample
`num_motif`	283	Motif features per region
`embed_dim`	768	Hidden dimension
`num_layers`	12	Number of DiT blocks
`num_heads`	12	Attention heads
`dropout`	0.1	Dropout rate
`mask_ratio`	0.5	50% of regions masked
`batch_size`	8	Training batch size
`lr`	0.0001	Learning rate
`use_lora`	true	LoRA fine-tuning enabled

4 Configuration Parameters

5 Layer Details

5.1 TimestepEmbedder

Embeds scalar timesteps into vector representations:

1. Sinusoidal positional embedding: $t \to \mathbb{R}^{256}$
2. MLP: Linear($256 \to 768$) + SiLU + Linear($768 \to 768$)
3. Output: conditioning vector $c \in \mathbb{R}^{768}$

5.2 DiTBlock

Each of the 12 DiT blocks contains:

Component	Details
`norm1`	LayerNorm(768), no affine parameters
`attn`	Self-Attention: qkv Linear($768 \to 2304$), 12 heads, proj Linear($768 \to 768$)
`norm2`	LayerNorm(768), no affine parameters
`mlp`	fc1: Linear($768 \to 3072$), GELU, fc2: Linear($3072 \to 768$)
`adaLN_modulation`	SiLU + Linear($768 \to 4608$) producing 6 vectors of dim 768

6 Training Flow

1. Load pretrained checkpoint (`checkpoint-799.pth`) with weight renaming
2. Apply LoRA to `region_embed` and `encoder` layers
3. For each batch:
   (a) Sample random timestep $t \in [0, 1000)$
   (b) Mask 50% of regions
   (c) Forward pass through DiT
   (d) Compute MSE loss on masked positions only
   (e) Backpropagation + optimizer step
4. Metrics: Pearson correlation, MSE, R^2 on masked predictions

7 Loss Function

The model uses MSE loss computed only on masked positions:

$$\mathcal{L} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \|\hat{x}_i - x_i\|^2$$

where $\mathcal{M}$ is the set of masked region indices, $\hat{x}_i$ is the predicted motif vector, and x_i is the ground truth.