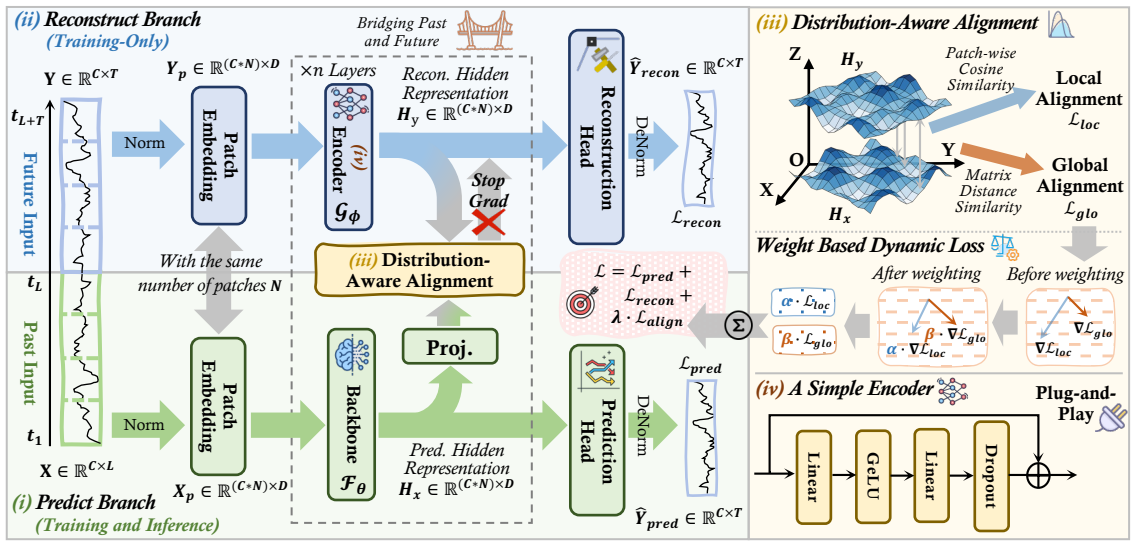
(ii) Reconstruct Branch
(Training-Only)
Bridging Past and Future
Y ∈ ℝ^{C×T}
Y_p ∈ ℝ^{(C*N)×D}
×n Layers
Recon. Hidden Representation
H_y ∈ ℝ^{(C*N)×D}
Reconstruction Head
Ŷ_recon ∈ ℝ^{C×T}
DeNorm
L_recon
t_{L+T}
Future Input
Norm
Patch Embedding
(iv) Encoder G_ϕ
Stop Grad
With the same number of patches N
(iii) Distribution-Aware Alignment
t_L
Past Input
t_1
X ∈ ℝ^{C×L}
Norm
Patch Embedding
X_p ∈ ℝ^{(C*N)×D}
Backbone F_θ
Proj.
Pred. Hidden Representation
H_x ∈ ℝ^{(C*N)×D}
Prediction Head
DeNorm
L_pred
Ŷ_pred ∈ ℝ^{C×T}
(i) Predict Branch
(Training and Inference)
L = L_pred + L_recon + λ · L_align
Σ
(iii) Distribution-Aware Alignment
Z
H_y
Patch-wise Cosine Similarity
Local Alignment
L_loc
O
Y
X
H_x
Matrix Distance Similarity
Global Alignment
L_glo
Weight Based Dynamic Loss
After weighting
Before weighting
α · L_loc
β · L_glo
α · ∇L_loc
β · ∇L_glo
∇L_glo
∇L_loc
(iv) A Simple Encoder
Plug-and-Play
Linear
GeLU
Linear
Dropout