A Repainting-based WaveStitch

To condition on both auxiliary features and available signal values during inference, we implement a RePainting-based version of `WaveStitch`, detailed in Algorithm 3. First, we apply a conditional mask to identify the known signal values, which is used to generate noised versions of the signal (step 1, Figure 8). The denoising process is then applied to the entire signal, with conditions re-introduced at each iteration to adjust the unconstrained parts of the signal in alignment with the known values.

We parallelize the generation of time series segments using overlapping windows of length w with stride s (Algorithm 3, line 2). Each mini-batch of windows is denoised in parallel, conditioned on the historical signals and auxiliary features. To enforce coherence across windows, the repainting-based stitching mechanism simply **overwrites** overlapping regions in each window with the corresponding region from the preceding window, gradually aligning non-overlapping parts in subsequent iterations (see Figure 9, line 14). This approach, ensures that coherence emerges progressively through iterative refinement.

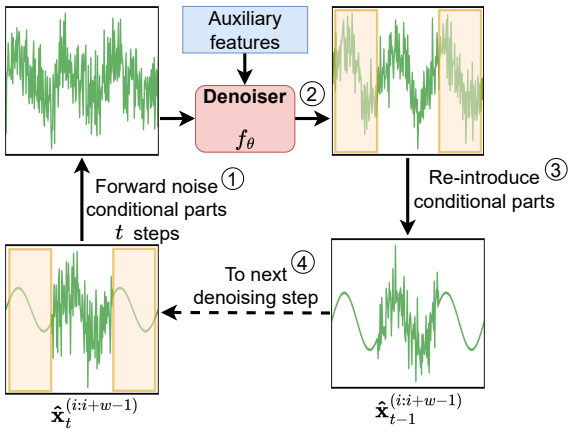

Figure 8: Conditional Denoising. Conditional values in orange.

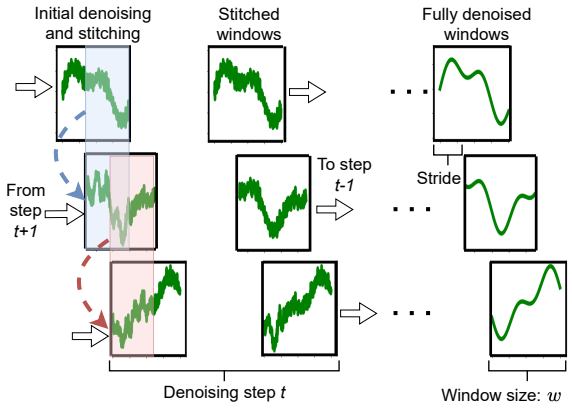

Figure 9: Parallel Denoising with Overlap Stitching

Algorithm 3 WaveStitch Inference with Repainting-based Conditioning

1: **Input:** Test time series data $\mathcal{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{M}$, where $\forall i, \mathbf{x}^{(i)} \in \mathbb{R}^{1\times C}$; Window size w; Stride s; mini-batch size b; Diffusion parameters $\{\alpha_t, \beta_t, \bar{\alpha}_t\}$; auxiliary features $\mathcal{A} = \{\mathbf{a}^{(i)}\}_{i=1}^{M}$; Conditional mask set $\mathcal{M} = \{\mathbf{m}^{(i)}\}_{i=1}^{M}$, where $\mathbf{m}^{(i)} = 1$ means the corresponding $\mathbf{x}^{(i)}$ is missing, and $\mathbf{m}^{(i)} = 0$ means $\mathbf{x}^{(i)}$ is conditional; Unconditional denoiser f_θ.

2: **Create windowed samples:**

$$\mathcal{X}_w = \{\mathbf{x}_w^{(i)} = \mathbf{x}^{(i.s-s+1:i.s-s+w)}\}_{i=1}^{\lfloor (M-w)/s \rfloor},$$
$$\mathcal{A}_w = \{\mathbf{a}_w^{(i)} = \mathbf{a}^{(i.s-s+1:i.s-s+w)}\}_{i=1}^{\lfloor (M-w)/s \rfloor},$$
$$\mathcal{M}_w = \{\mathbf{m}_w^{(i)} = \mathbf{m}^{(i.s-s+1:i.s-s+w)}\}_{i=1}^{\lfloor (M-w)/s \rfloor},$$

3: **Initialize output windows:**

$$\hat{\mathcal{X}}_w = \{\hat{\mathbf{x}}_w^{(i)} = (1-\mathbf{m}_w^{(i)}) \cdot \mathbf{x}_w^{(i)} + \mathbf{m}_w^{(i)} \cdot \mathbf{z}^{(i)}\}_{i=1}^{\lfloor (M-w)/s \rfloor}$$

, where $\mathbf{z}^{(i)} \sim \mathcal{N}(0, I_{w\times C})$

4: **Divide $\mathcal{X}_w, \mathcal{A}_w, \mathcal{M}_w$ into $(M-w)/(b\times s)$ mini-batches**

5: **for** each mini-batch **do**

6: **for** each timestep $t = T, T-1, \ldots, 1$ **do**

7: **if** $t = T$ **then**

$$\hat{\mathbf{x}}_{w,t}^{(i)} = \hat{\mathbf{x}}_w^{(i)}$$

8: **for** $(\mathbf{x}_w^{(i)}, \mathbf{a}_w^{(i)}, \mathbf{m}_w^{(i)})$ in mini-batch in **parallel**: **do**

9: **Conditional Forward noising**:

$$\hat{\mathbf{x}}_{w,t}^{(i)} = (1-\mathbf{m}_w^{(i)}) \cdot (\sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_w^{(i)} + \sqrt{1-\bar{\alpha}_t} \cdot \epsilon^{(i)}) + \mathbf{m}_w^{(i)} \cdot \hat{\mathbf{x}}_{w,t}^{(i)}$$

10: **One-step denoising**:

$$\hat{\mathbf{x}}_{w,t-1}^{(i)} = \frac{1}{\sqrt{\alpha_t}} \left(\hat{\mathbf{x}}_{w,t}^{(i)} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \cdot f_\theta \left(\mathbf{a}_w^{(i)}, \hat{\mathbf{x}}_{w,t}^{(i)}, t \right) \right)$$

11: **Re-introduce conditions:**

$$\hat{\mathbf{x}}_{w,t-1}^{(i)} = (1-\mathbf{m}_w^{(i)}) \cdot \mathbf{x}_w^{(i)} + \mathbf{m}_w^{(i)} \cdot \hat{\mathbf{x}}_{w,t-1}^{(i)}$$

12: **if** $s < w$ **then**

13: **for** $i > 1$ in **parallel do**

14: **Stitch overlaps:**

$$\hat{\mathbf{x}}_w^{(i)(1:w-s)} = \hat{\mathbf{x}}_w^{(i-1)(1+s:w)}$$

15: **Merge windows:**

$$\hat{\mathcal{X}} = \hat{\mathbf{x}}_w^{(1)} \cup \left(\bigcup_{i\geq 2} \hat{\mathbf{x}}_w^{(i)(w-s+1:w)} \right)$$

16: **return** $\hat{\mathcal{X}}$