# Deep Parameter Interpolation for Scalar Conditioning

Supplementary Material

## 7. Learnable vs. Fixed Interpolation Functions

A central component of deep parameter interpolation (DPI) is the learnable monotonic interpolation function $\lambda(s)$, which determines how the model transitions between the two parameter sets across the scalar domain. While Figure 3 illustrates that $\lambda(s)$ adapts its shape depending on the architecture and generative framework, we additionally assess whether this learned flexibility is necessary or whether a fixed, non-learnable interpolation rule would be sufficient.

To evaluate this, we compare DPI with a fixed linear monotonic interpolation

$$\lambda(s) = s, \quad s \in [0, 1],$$

which removes all learnable parameters from the interpolation function while keeping all other aspects of DPI unchanged.

We train DRUNet and ADM architectures under both diffusion and flow-matching settings using exactly the same configurations described in Section 4.1. For each model, we compute the corresponding training objective on 500 held-out images across the whole scalar range (i.e., 1,000 scalar steps). At each scalar, we evaluate 20 noise realizations using shared random seeds to ensure fair comparison.

Table 5 reports the averaged diffusion and flow objectives. In every setting, the learnable interpolation function achieves strictly lower error than the fixed linear rule, demonstrating that the ability to adapt $\lambda(s)$ is beneficial even though both versions interpolate between identical parameter endpoints.

Table 5. Objective comparison between fixed linear and learnable interpolation functions. The learnable monotonic interpolation consistently reduces the diffusion and flow objectives compared to a non-learnable linear function. **Best** results are color-coded per sampling methods and per-architecture.

| | | **Diffusion Objective** | **Flow Objective** |
|---|---|---|---|
| **Method** | $\lambda(s)$ | $\mathbb{E}\lVert\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t\,;\sigma_t) - \boldsymbol{\epsilon}\rVert_2^2$ | $\mathbb{E}\lVert\boldsymbol{v}_\theta(\boldsymbol{x}_t\,;t) - \boldsymbol{v}\rVert_2^2$ |
| **DRUNet** | $s$ | 1.870e−2 | 1.250e−1 |
| | Learnable | **1.865e−2** | **1.209e−1** |
| **ADM** | $s$ | 1.775e−2 | 1.191e−1 |
| | Learnable | **1.772e−2** | **1.086e−1** |