Beyond Optimal Transport: Model-Aligned Coupling for Flow Matching

Supplementary Material

7. Proof of Theorem 1

In this section, we provide the complete proof for the curvature bound presented in the main text.

7.1. Preliminaries and Definitions

Recall the decomposition of the learned velocity field:

$$v_\theta(x_t, t) = d + e(x_t, t), \tag{9}$$

where $d = x_1 - x_0$ is the constant ground-truth transport direction, and $e(x_t, t)$ is the prediction error (perturbation). The trajectory evolves according to the ODE $\dot{x}_t = v_\theta(x_t, t)$.

We assume the following bounds on the perturbation $e(x_t, t)$ for all $t \in [0, 1]$:

$$\|e(x_t, t)\| \le \varepsilon < \|d\|, \tag{10}$$

$$\|\nabla_x e(x_t, t)\|_2 \le L_x, \tag{11}$$

$$\|\partial_t e(x_t, t)\| \le L_t, \tag{12}$$

where $\|\cdot\|$ denotes the Euclidean norm for vectors, and $\|\cdot\|_2$ denotes the induced spectral norm (operator norm) for matrices.

The curvature $K(t)$ of a parameterized curve $x(t)$ in Euclidean space is defined as the magnitude of the normal acceleration normalized by the square of the velocity:

$$K(t) = \frac{\|a_\perp(t)\|}{\|v_\theta(x_t, t)\|^2}, \tag{13}$$

where $a(t) := \ddot{x}_t = \dot{v}_\theta(x_t, t)$ is the total acceleration, and $a_\perp(t)$ is its component orthogonal to the velocity direction. Note that by definition, $\|a_\perp(t)\| \le \|a(t)\|$.

7.2. Derivation of the Bound

Step 1: Acceleration Bound. First, we derive the expression for the total acceleration $a(t)$ using the material derivative (chain rule). Since d is constant, we have $\nabla_x d = 0$ and $\partial_t d = 0$. Thus:

$$\begin{aligned} a(t) &= \frac{\mathrm{d}}{\mathrm{d}t} v_\theta(x_t, t) \\ &= \partial_t v_\theta(x_t, t) + (\nabla_x v_\theta(x_t, t)) \cdot \dot{x}_t \\ &= \partial_t e(x_t, t) + (\nabla_x e(x_t, t)) \cdot v_\theta(x_t, t) \\ &= \partial_t e(x_t, t) + (\nabla_x e(x_t, t))(d + e(x_t, t)). \end{aligned} \tag{14}$$

Taking the norm on both sides and applying the triangle inequality and the definition of the operator norm ($\|Ax\| \le \|A\|_2 \|x\|$):

$$\begin{aligned} \|a(t)\| &= \|\partial_t e(x_t, t) + (\nabla_x e(x_t, t))(d + e(x_t, t))\| \\ &\le \|\partial_t e(x_t, t)\| + \|\nabla_x e(x_t, t)\|_2 \cdot \|d + e(x_t, t)\| \\ &\le \|\partial_t e(x_t, t)\| + \|\nabla_x e(x_t, t)\|_2 \left(\|d\| + \|e(x_t, t)\|\right). \end{aligned} \tag{15}$$

Substituting the assumptions from Eqs. (10)–(12), we obtain an upper bound on the acceleration magnitude:

$$\|a(t)\| \le L_t + L_x(\|d\| + \varepsilon). \tag{16}$$

Step 2: Velocity Lower Bound. Next, we establish a lower bound for the velocity magnitude. Using the reverse triangle inequality:

$$\begin{aligned} \|v_\theta(x_t, t)\| &= \|d + e(x_t, t)\| \\ &\ge \|d\| - \|e(x_t, t)\| \\ &\ge \|d\| - \varepsilon. \end{aligned} \tag{17}$$

Since we assumed $\varepsilon < \|d\|$, it follows that $\|v_\theta(x_t, t)\| > 0$, ensuring the curvature is well-defined. Squaring this term gives the lower bound for the denominator:

$$\|v_\theta(x_t, t)\|^2 \ge (\|d\| - \varepsilon)^2. \tag{18}$$

Step 3: Curvature Bound. Finally, substituting the bounds from Eq. (16) and Eq. (18) into the definition of curvature:

$$\begin{aligned} K(t) &= \frac{\|a_\perp(t)\|}{\|v_\theta(x_t, t)\|^2} \\ &\le \frac{\|a(t)\|}{\|v_\theta(x_t, t)\|^2} \\ &\le \frac{L_t + L_x(\|d\| + \varepsilon)}{(\|d\| - \varepsilon)^2}. \end{aligned} \tag{19}$$

This concludes the proof of the Theorem. □

8. More Implementation Details

Network Structure We employ different backbone architectures tailored to the resolution and complexity of the datasets. For high-resolution generation tasks (AFHQ-v2 and CelebA-HQ-256), we utilize the Diffusion Transformer (DiT) [17]. For the lower-resolution CIFAR-10 dataset, we adopt a U-Net backbone [3].

DiT Architecture (AFHQ-v2 & CelebA-HQ-256). For both AFHQ-v2 and CelebA-HQ-256, we adopt the **DiT-B/2** configuration. The models operate on a latent representation with a patch size of 2. The detailed configurations are summarized in Tab. 3. Both models utilize the standard Transformer backbone composed of multi-head self-attention and point-wise feed-forward networks with AdaLN conditioning.

Table 3. DiT model configuration for AFHQ-v2 and CelebA-HQ-256. Both datasets share the identical DiT-B/2 architecture for unconditional generation.

Hyperparameter	Value
Input Shape	$(4 \times 32 \times 32)$
Patch Size	2
Depth (Layers)	12
Hidden Size	768
Number of Heads	12
Positional Encoding	2D Sin-Cos
Input Embedding	PatchEmbed

U-Net Architecture (CIFAR-10). For CIFAR-10 (32×32), we employ a U-Net architecture enhanced with attention mechanisms, following previous work [30]. The model is constructed with a wide ResNet backbone and utilizes spatial self-attention at lower resolutions. Specifically, we use a base channel width of 128 with channel multipliers of $[1, 2, 2, 2]$, resulting in a deep hierarchy. Detailed hyperparameters derived from our implementation are listed in Tab. 4.

Table 4. U-Net model configuration used for the CIFAR-10 dataset.

Hyperparameter	Value
Input Shape	$(3 \times 32 \times 32)$
Base Channels	128
Channel Multipliers	$(1, 2, 2, 2)$
ResBlocks per Scale	2
Attention Resolution	16×16
Attention Head Channels	64
Dropout	0.1

8.1. Detailed Implementation with Different Base Models

In this section, we provide detailed implementation algorithms for integrating Model-Aligned Coupling (MAC) with four representative generative models: Flow Matching, BatchOT, Shortcut Models, and MeanFlow. For all methods, MAC introduces an reweighting mechanism based on the learnability of coupling pairs (x_0, x_1), where $x_0 \sim p_0$ and $x_1 \sim p_1$.

MAC with Flow Matching (FM) Flow Matching [13] is a fundamental framework. It regresses a velocity field $v_\theta(x, t)$ towards a conditional vector field $u_t(x) = x_1 - x_0$. The prediction error for MAC is computed using the standard velocity field evaluation at the endpoints:

$$\widehat{\mathcal{L}}_{\text{pair}}(x_0, x_1) = \frac{1}{2}\Big(\|v_\theta(x_0, 0) - (x_1 - x_0)\|^2 + \|v_\theta(x_1, 1) - (x_1 - x_0)\|^2\Big). \quad (20)$$

We calculate this error for each pair in the batch to obtain weights w_i (see Eq. 8 in the main paper), which are then applied to the standard Flow Matching loss. The pseudocode is shown on Algo. 2.

Algorithm 2 MAC with Flow Matching

Require: Model v_θ, batch size B
1: **for** each training iteration **do**
2: Sample data $x_1 \sim p_1(x)$ and noise $x_0 \sim p_0(x)$
3: **MAC Step:** Compute prediction errors via Eq. 20 and derive weights $W = \{w_i\}_{i=1}^B$
4: Sample time $t \sim \mathcal{U}[0, 1]$
5: Interpolate $x_t = (1 - t)x_0 + tx_1$ and target $v = x_1 - x_0$
6: Compute weighted loss: $\mathcal{L} = \frac{1}{B}\sum_{i=1}^B w_i \|v_\theta(x_t^{(i)}, t) - v^{(i)}\|^2$
7: Update $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}$
8: **end for**

MAC with BatchOT BatchOT [18] utilizes Optimal Transport to couple samples within a mini-batch. Following the original setting, we treat the OT plan γ as a joint probability distribution and resample pairs (x_0, x_1) from it. Since BatchOT uses the same model architecture as standard Flow Matching, the prediction error is identical to Eq. 20. The pseudocode for BatchOT-MAC is shown on Algo. 3.

MAC with Shortcut Models Shortcut Models [5] parameterize the velocity field $v_\theta(x, t, d)$ with an additional input d representing the step size. To estimate the learnability of a trajectory, we evaluate the model at the step size $d = 0$. The prediction error is formulated as:

$$\widehat{\mathcal{L}}_{\text{pair}}(x_0, x_1) = \tfrac{1}{2}\Big(\|v_\theta(x_0, 0, 0) - (x_1 - x_0)\|^2 + \|v_\theta(x_1, 1, 0) - (x_1 - x_0)\|^2\Big). \quad (21)$$

Algorithm 3 MAC with BatchOT

Require: Model v_θ, batch size B, Sinkhorn reg ε

for each training iteration **do**
  Sample mini-batch $X_1 \sim p_1(x)$, $X_0 \sim p_0(x)$
  Compute cost matrix $M_{ij} = \|X_0^{(i)} - X_1^{(j)}\|^2$
  Solve OT plan $\gamma = \text{Sinkhorn}(M, \varepsilon)$
  OT Resampling: Sample indices $(i, j) \sim \gamma$ to form a new paired batch $\{(X_0^{(i)}, X_1^{(j)})\}_{k=1}^B$
  MAC Step: Compute weights W for these resampled pairs using Eq. 20
  Proceed with weighted Flow Matching (lines 4-7 of Alg. 2)
end for

The weights w_i derived from this error are applied to the Shortcut loss $\mathcal{L}_{\text{SC}}$, which combines a standard flow matching loss (at $d = 0$) with a self-consistency constraint (at $d > 0$). The Shortcut loss for is defined as:

$$\mathcal{L}_{\text{SC}} = \underbrace{\|v_\theta(x_t, t, 0) - (x_1 - x_0)\|^2}_{\text{Flow Matching}} + \underbrace{\|v_\theta(x_t, t, 2d) - v_{\text{target}}\|^2}_{\text{Self Consistency}}, \tag{22}$$

where $x_t = (1-t)x_0 + tx_1$. The bootstrap target v_{target} is constructed by averaging two consecutive steps of size d:

$$v_{\text{target}} = \frac{1}{2} v_\theta(x_t, t, d) + \frac{1}{2} v_\theta(x'_{t+d}, t+d, d),$$

with the intermediate state computed as $x'_{t+d} = x_t + d \cdot v_\theta(x_t, t, d)$.

Algorithm 4 MAC with Shortcut Models

Require: Model $v_\theta(\cdot, \cdot, d)$, batch size B

for each training iteration **do**
  Sample $x_1 \sim p_1(x)$ and $x_0 \sim p_0(x)$
  MAC Step: Compute prediction errors with $d = 0$ via Eq. 21 and derive weights W
  Prepare training samples:
    - Flow Matching samples ($d = 0$) with x_t
    - Self-Consistency bootstrap samples ($d > 0$) with targets v_{target}
  Compute per-sample combined loss $\mathcal{L}_{\text{SC}}^{(i)}$
  Compute weighted loss: $\mathcal{L} = \frac{1}{B} \sum_{i=1}^B w_i \mathcal{L}_{\text{SC}}^{(i)}$
  Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
end for

MAC with MeanFlow MeanFlow [6] learns an average velocity field $v_\theta(x_t, t, r)$ that represents the transport between time t and r. The model is governed by the integral equation $(r-t)v_\theta(x_t, t, r) = \int_t^r v(x_\tau, \tau) d\tau$, where $v(x_\tau, \tau)$ is the instantaneous ground-truth velocity. Differentiating this condition with respect to t yields the training objective:

$$v_\theta(x_t, t, r) = \text{sg}\left((r-t) \frac{d}{dt} v_\theta(x_t, t, r) + v(x_t, t) \right), \tag{23}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operation.

To estimate the learnability of a trajectory for MAC, we evaluate the model at the limit where the time horizon vanishes, i.e., setting $r = t$. The prediction error is computed at the endpoints:

$$\widehat{\mathcal{L}}_{\text{pair}}(x_0, x_1) = \tfrac{1}{2}\Big(\|v_\theta(x_0, 0, 0) - (x_1 - x_0)\|^2 + \|v_\theta(x_1, 1, 1) - (x_1 - x_0)\|^2 \Big). \tag{24}$$

We compute these errors to derive weights w_i, which are then applied to the MeanFlow consistency loss.

Algorithm 5 MAC with MeanFlow

Require: Model $v_\theta(\cdot, \cdot, r)$, batch size B

for each training iteration **do**
  Sample pairs (x_0, x_1) with $x_0 \sim p_0, x_1 \sim p_1$
  MAC Step: Compute prediction errors with $r = t$ via Eq. 24 and derive weights W
  Sample time points t, r (e.g., via Logit-Normal sampling)
  Interpolate $x_t = (1-t)x_0 + tx_1$
  Compute derivative term $\dot{v}_\theta = \frac{d}{dt} v_\theta(x_t, t, r)$ (e.g., via JVP)
  Construct target: $v_{\text{tgt}} = (r-t)\dot{v}_\theta + (x_1 - x_0)$
  Compute weighted loss: $\mathcal{L} = \frac{1}{B} \sum_{i=1}^B w_i \|v_\theta(x_t^{(i)}, t, r) - \text{sg}(v_{\text{tgt}}^{(i)})\|^2$
  Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
end for

9. Computational Overhead

MAC does not introduce additional model parameters and inference overhead. Its only extra cost during training is one additional forward pass used to estimate prediction errors for coupling selection. On CIFAR-10 with MeanFlow, the training throughput is 0.2101 s/it with MAC, compared with 0.1742 s/it for the baseline, indicating a modest training-time overhead in exchange for improved few-step generation performance.

10. Usage Guidance for MAC

MAC is most beneficial in the few-step regime, where the quality of the learned transport path has the largest impact on generation performance. In the many-step regime, the

Table 5. One-step, four-step, and full-step (128 steps) FID scores on CIFAR-10 (class-conditional), CIFAR-10 (unconditional), AFHQ-v2 (unconditional), and CelebA-HQ-256 (unconditional). Lower is better.

Method	CIFAR-10 (Cond.)			CIFAR-10 (Uncond.)			AFHQ-v2 (Uncond.)			CelebA-HQ (Uncond.)		
	1-Step	4-Step	128-Step	1-Step	4-Step	128-Step	1-Step	4-Step	128-Step	1-Step	4-Step	128-Step
Flow Matching	290.30	30.23	7.11	353.18	52.40	7.30	365.86	57.36	7.56	287.25	65.83	7.36
Flow Matching + MAC	286.11	29.06	7.03	352.11	52.06	7.20	360.49	54.13	7.85	283.87	63.32	7.92
Flow Matching + MAC-full	276.79	29.52	6.63	344.19	49.49	7.51	360.06	52.41	7.40	279.85	63.85	7.14
BatchOT	210.60	21.33	4.86	293.91	31.28	7.14	282.14	45.95	7.82	225.68	54.26	6.75
BatchOT + MAC	207.56	21.14	4.97	289.20	30.69	7.84	284.28	43.05	8.19	219.76	51.49	6.79
BatchOT + MAC-full	208.37	20.89	5.03	289.79	30.92	7.45	281.05	41.52	7.57	224.75	52.84	6.67
Shortcut	22.26	20.45	6.77	30.46	21.30	8.39	27.41	11.23	7.38	22.83	15.29	7.64
Shortcut + MAC	20.56	18.79	6.97	26.70	17.77	8.43	26.78	10.86	8.08	21.98	14.31	7.89
Shortcut + MAC-full	20.52	17.79	6.65	29.47	18.41	8.41	26.83	10.44	7.46	20.38	12.63	7.24
MeanFlow	18.27	11.91	7.73	23.80	13.99	9.54	14.42	8.17	8.61	9.39	8.47	7.61
MeanFlow + MAC	18.04	11.86	7.19	22.99	13.71	8.60	13.47	7.86	8.85	9.18	8.36	7.74
MeanFlow + MAC-full	18.12	11.43	6.99	22.41	13.16	9.34	13.79	7.89	8.17	9.12	7.76	6.37

Table 6. FID (Euler / Heun) on CIFAR-10 (unconditional).

Method	1-step	4-step	128-step
MeanFlow	23.80 / 23.80	13.99 / 23.64	9.54 / 9.20
MeanFlow + MAC	**22.99 / 22.99**	**13.71 / 21.87**	**8.60 / 8.64**

baseline already has sufficient integration steps to follow the learned flow accurately, so the benefit of improved coupling construction naturally becomes smaller. Therefore, we recommend enabling MAC when the goal is to improve one-step or few-step generation, while its advantage may be less pronounced when the number of sampling steps is large.

11. More Results

Full Coupling Optimization. In the main paper, we adopt a top-k selection strategy to approximate the ideal coupling $\tilde{\rho}$ defined in Eq. 5, by selecting a fixed fraction of source-target couplings with the lowest prediction error. Here, we investigate a more accurate alternative that performs *full coupling optimization* by solving Eq. 5 over all possible source-target couplings within each batch. Specifically, we solve the full assignment problem defined in Eq. 5 by applying the Sinkhorn algorithm to minimize the expected pairwise prediction error over all source-target couplings. We refer to this variant as MAC-full.

As shown in Tab. 5, the full coupling variant consistently achieves better performance across datasets and sampling steps, highlighting the benefits of more precise pairwise supervision.

Evaluation with Different Solvers In the main paper, we use Euler sampling following prior work. To verify that the effectiveness of MAC is not tied to a specific numerical solver, we additionally evaluate it with both Euler and Heun on CIFAR-10. The results are reported in Table 6. MAC consistently improves MeanFlow under both solvers. Note that in the 1-step setting, Heun reduces to Euler, since there is no subsequent step at which a corrected slope can be evaluated. In the few-step regime, Heun can be less effective because MeanFlow predicts an *averaged* velocity over $[t, t+h]$, whereas Heun assumes an *instantaneous* velocity and mixes directions across adjacent intervals when h is large. In the 128-step regime, where h becomes small, this mismatch diminishes and Heun can slightly outperform Euler.

Additional Qualitative Results To further validate the effectiveness and generality of our proposed MAC strategy, we provide qualitative comparisons of generated samples across multiple base flow-matching models and their MAC-augmented variants. Specifically, we visualize samples from the following models: Flow Matching, Flow Matching + MAC, BatchOT, BatchOT + MAC, Shortcut, Shortcut + MAC, MeanFlow, and MeanFlow + MAC. For each model, we present generated samples under three sampling settings: one-step, four-step, and full (128-step) generation.

Fig. 4 shows qualitative visualizations on AFHQ-v2 and CelebA-HQ-256. Across both datasets, MAC can improve visual fidelity compared to the corresponding base models.

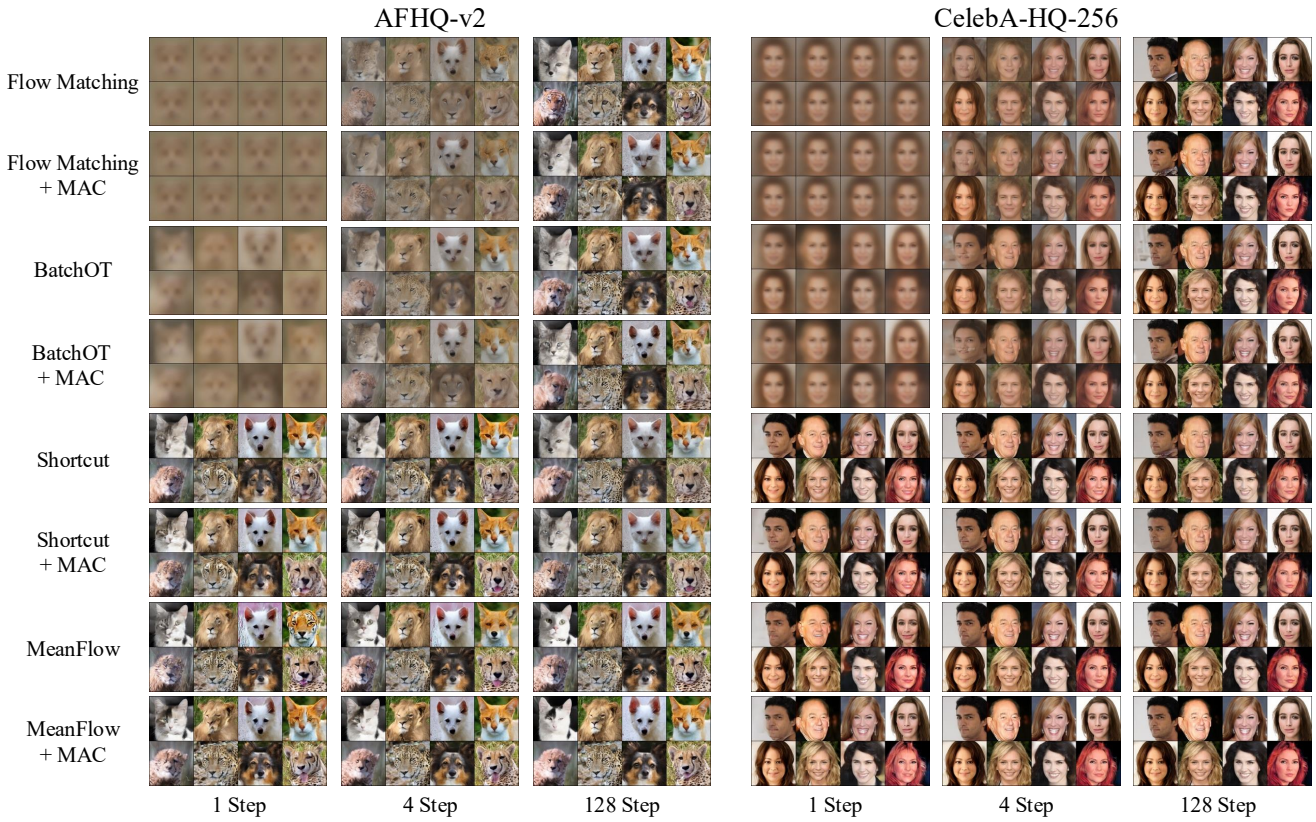

Figure 4. Qualitative comparison of generated AFHQ-v2 and CelebA-HQ-256 samples across different models under varying sampling steps (1, 4, and 128 steps). Methods with MAC generate better images, especially in the one-step and four-step settings.