Project Aether: Latent Space Transport Control

A Unified Framework for Safe Concept Steering in Generative AI

Technical Architecture Document
Version 3.1 (With Layer Sensitivity Analysis)

November 2025

Abstract

This document details the "Latent Space Transport Control" framework. We address the alignment problem in generative models by learning a policy π_ϕ to steer latent trajectories. This approach builds upon the "Probability Flow ODE" formulation established in MIT 6.S184 (2025) and adapts the latent reinforcement learning techniques proposed by Wagenmaker et al. (2025). The goal is to maximize safety (Lamba et al., 2025) while minimizing transport cost.

Key Contributions:

- Unified ODE framework for diffusion and flow-matching models
- Linear probing for concept detection in latent space
- **Layer sensitivity analysis** to identify optimal intervention points
- Optimal transport reward combining safety and semantic alignment
- Evaluation framework with SSR, LPIPS, and transport cost metrics

Contents

1 Part I: The Conceptual Vision

1.1 The Definition of Generative Modeling

To understand our control mechanism, we must first define the process we are controlling.

Reference Source: Song et al. [30] via MIT 6.S184

As concisely defined in arXiv:2506.02070v2:

> *"Creating noise from data is easy; creating data from noise is generative modeling."*

Our project focuses exclusively on the second half: controlling the trajectory as we transform noise back into data.

1.2 The Alignment Gap

Modern Generative AI models (Diffusion and Flow Matching) are powerful but lack fine-grained control over "safety concepts."

Reference Source: Lamba et al. (2025) - Reference 1

As surveyed in *Alignment and Safety of Diffusion Models via Reinforcement Learning*, traditional methods like Supervised Fine-Tuning (SFT) are computationally expensive and can degrade general model performance. Lamba et al. highlight that Reinforcement Learning (RL) offers a promising alternative by optimizing a reward function (Safety) without needing a paired dataset of "Safe/Unsafe" images.

1.3 The Solution: The "Autopilot" Analogy

We are not retraining the artist. We are building an **Autopilot** that steers the generation process.

Intuition: The River and the Boat

Imagine image generation as a boat floating down a river.

- **Upstream** ($T = 1$): Pure Static/Noise.
- **Downstream** ($T = 0$): A clear, finished photograph.
- **The Current** (v_θ): The AI model's learned velocity field.

Our Intervention: We attach a rudder (Policy π) to the boat. If the current pulls toward the "Unsafe Waterfall," the policy applies a nudge (Δz) to steer into a safe channel.

2 Part II: The Theoretical Engine

We formulate the generation process using Stochastic Differential Equations (SDEs) and Vector Fields.

2.1 The Probability Flow ODE

While Diffusion models are stochastic, for control purposes we utilize the deterministic Probability Flow ODE.

Reference Source: Holderrieth & Erives (2025) - MIT 6.S184

This formulation is derived directly from MIT Class 6.S184: *Generative AI With Stochastic Differential Equations*, specifically Section 3.1: Conditional and Marginal Probability Path.
The course defines the generative process not just as denoising, but as following a **Vector Field** over time.

Formal Definition: The Marginal Vector Field

The trajectory is governed by the ODE:

$$\frac{dz}{dt} = v_\theta(z_t, t) \tag{1}$$

According to MIT 6.S184 (Section 3.2), the optimal vector field $u_t(x)$ satisfies:

$$u_t(x) = \mathbb{E}[u_t(x|x_1) \,|\, x_t = x]$$

Our base model v_θ is a neural network approximation of this vector field.
In standard generation, we solve this via numerical integration (e.g., Euler method):

$$z_{\text{next}} \approx z_t + v_\theta(z_t, t) \cdot \Delta t$$

2.2 Why Deterministic?

We choose the ODE formulation over the SDE formulation because it provides a **unique, reversible mapping** between the latent space z_T and the image space z_0. This is crucial for RL stability—if the environment is random (SDE), the agent struggles to learn cause-and-effect.

3 Part II.5: Layer Sensitivity Analysis

We now address a critical question from our research agenda: **which latent layers (or timesteps) are most effective for policy intervention?**

3.1 Motivation: The Layer Selection Problem

Not all layers in a generative model are equally amenable to steering. Intervening too early (high t) may disrupt global structure, while intervening too late (low t) may be ineffective as the trajectory has already committed to a concept.

Reference Source: Alain & Bengio (2016)

The foundational work *Understanding Intermediate Layers Using Linear Classifier Probes* establishes that linear probes can measure the "immediate suitability" of representations for classification at each layer. We adapt this methodology to identify optimal intervention points in generative models.

Reference Source: Lu et al. (2024) - MACE

The MACE paper introduces the concept of **Semantic Single Boundary (SSB)**: *"After the turning point (SSB), the specific mode to be fully denoised is determined."* This suggests that:

- Interventions **before** SSB affect all concepts (global effect)
- Interventions **after** SSB can be concept-specific (local effect)

The optimal intervention point is typically **at or near the SSB**.

3.2 The Layer Sensitivity Score

We define a **Layer Sensitivity Score** S_ℓ for each layer (or timestep) ℓ that combines three factors:

$$S_\ell = \underbrace{\text{Acc}_\ell}_{\text{Probe Accuracy}} \times \underbrace{(1 - \text{FID}_\ell^{\text{norm}})}_{\text{Quality Preservation}} \times \underbrace{\Delta\text{SSR}_\ell}_{\text{Steering Effectiveness}} \tag{2}$$

Component Definitions:

- Acc_ℓ: Linear probe accuracy at layer ℓ for the safety concept. Higher accuracy indicates the concept is more linearly separable at this layer.
- $\text{FID}_\ell^{\text{norm}}$: Normalized FID (Fréchet Inception Distance) degradation when intervening at layer ℓ. Lower values indicate better quality preservation.
- ΔSSR_ℓ: Change in Safety Success Rate from intervention at layer ℓ. Higher values indicate more effective steering.

3.3 The Probing Protocol

Algorithm 1 Layer Sensitivity Analysis

Require: Frozen base model v_θ, concept classifier C, timesteps $\{t_1, \ldots, t_T\}$
Ensure: Optimal intervention layer ℓ^*

1: **for** each timestep $t_\ell \in \{t_1, \ldots, t_T\}$ **do**
2: Collect latent states $\{z_{t_\ell}^{(i)}\}_{i=1}^N$ from N trajectories
3: Train linear probe: $\hat{y} = \sigma(w_\ell \cdot z_{t_\ell} + b_\ell)$
4: Compute Acc_ℓ on held-out validation set
5: Apply steering at layer ℓ, measure ΔSSR_ℓ and FID_ℓ
6: Compute sensitivity score S_ℓ using Equation 2
7: **end for**
8: $\ell^* = \arg\max_\ell S_\ell$
9: **return** ℓ^*

3.4 Expected Findings

Based on the literature and our theoretical analysis, we hypothesize:

1. **Early layers** ($t \approx 1.0$): High noise level. Probe accuracy is low because representations are not yet semantically meaningful. Steering has diffuse, unpredictable effects.

2. **Middle layers** ($t \approx 0.4$–0.5): **OPTIMAL**. This is near the Semantic Single Boundary (SSB). Concepts are linearly separable (high probe accuracy) AND the trajectory is still malleable (high steering effectiveness).

3. **Late layers** ($t \approx 0.0$): High probe accuracy but low steering effectiveness. The trajectory has already committed to a specific concept/mode. Interventions here cannot significantly alter the output.

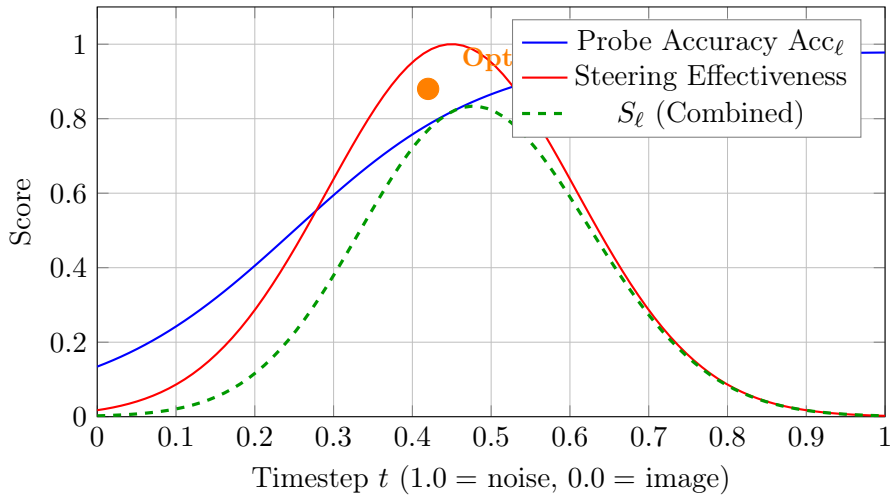

Figure 1: Hypothesized Layer Sensitivity Curves. The optimal intervention point occurs at mid-timesteps ($t \approx 0.4$–0.5) where both probe accuracy and steering effectiveness are high. This corresponds to the Semantic Single Boundary (SSB) region.

3.5 Integration with Optimal Transport Reward

The layer sensitivity analysis informs our OT reward (Section 5) by identifying **where** to apply steering. We extend the reward function to be layer-weighted:

$$J(\phi) = \mathbb{E}\left[R_{\text{safe}}(x_0) - \lambda \sum_t \underbrace{w_t}_{\text{Layer Weight}} \|a_t\|^2\right] \tag{3}$$

where $w_t \propto S_t^{-1}$. This weighting scheme:

- **Penalizes more heavily** interventions at low-sensitivity layers (where steering is ineffective or harmful)
- **Encourages** the policy to focus its steering budget on high-sensitivity layers
- **Reduces** total transport cost while maintaining safety effectiveness

Key Insight

Layer sensitivity analysis transforms our steering problem from "how much to steer" to "where and how much to steer." This principled approach reduces total transport cost while maintaining or improving safety effectiveness.

4 Part III: The Steering Architecture

4.1 The Agent (The Pilot)

We introduce a lightweight neural network, the Policy Agent $\pi_\phi(z_t, t)$.

Reference Source: Wagenmaker et al. (2025) - Reference 5

This architecture is adapted from *Steering Your Diffusion Policy with Latent Space Reinforcement Learning* (arXiv:2506.15799v2).
Wagenmaker et al. demonstrate that:

> *"Latent space interventions are more sample efficient than pixel-space control."*

We apply their robotics-focused "Diffusion Policy" concept to image generation safety.

4.2 The Closed-Loop Control System

We intervene directly in the integration loop. The concept of modifying the latent code z to change attributes originates from GAN literature (Abbasian et al., 2023), but unlike GANs where the latent is static, our latent z_t evolves over time.

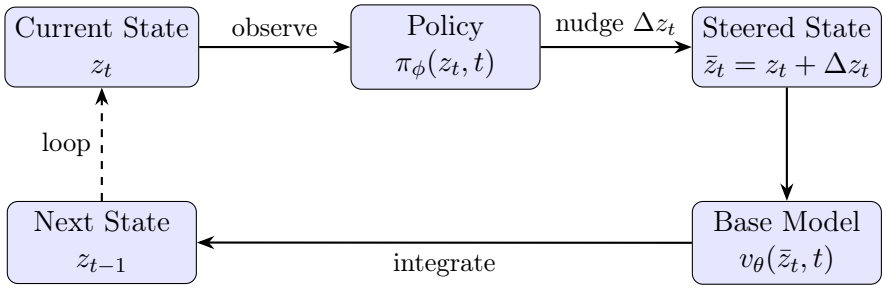

Figure 2: The Steering Loop. The policy intervenes **before** the ODE solver calculates the next step, allowing it to redirect the trajectory.

The complete update equation is:

$$\Delta z_t = \pi_\phi(z_t, t, \text{score}(z_t)) \quad \text{(Policy output)} \tag{4}$$

$$\bar{z}_t = z_t + \Delta z_t \quad \text{(Apply steering)} \tag{5}$$

$$z_{t-\Delta t} = \bar{z}_t - v_\theta(\bar{z}_t, t) \cdot \Delta t \quad \text{(ODE step)} \tag{6}$$

5 Part IV: Optimal Transport Reward

5.1 The Conflict

We must balance **Safety** (Reference 1) with **Semantic Fidelity** (Reference 3). If we only reward Safety, the agent will destroy the image to satisfy the classifier—a phenomenon known as **Reward Hacking**.

5.2 The Solution: Wasserstein Cost

We treat the steering problem as an **Optimal Transport** problem. We want to move the probability mass from the Unsafe distribution to the Safe distribution with the *minimum amount of work*.

Formal Definition: The Objective Function

The agent maximizes the following reward:

$$J(\phi) = \mathbb{E}\left[\underbrace{R_{\text{safe}}(x_0)}_{\text{Lamba et al.}} - \lambda \underbrace{\sum_t \|a_t\|^2}_{\text{Transport Cost}}\right] \tag{7}$$

where:

- R_{safe}: Output of a safety classifier (e.g., $+1$ if Safe, -1 if Unsafe).
- $\|a_t\|^2$: The squared norm of the steering action (energy cost).
- λ: Hyperparameter controlling the safety-quality tradeoff.

Theoretical Basis: Wasserstein-2 Distance

The term $\sum_t \|a_t\|^2$ is the **dynamic formulation of the Wasserstein-2 distance**. In optimal transport theory, the W_2 distance between distributions μ and ν is:

$$W_2(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \left(\int \|x - y\|^2 \, d\gamma(x, y) \right)^{1/2}$$

By minimizing the cumulative squared displacement, we ensure that the steered image x_{safe} is the "closest possible" image to the original x_{unsafe} on the data manifold.

5.3 The λ Tradeoff

The parameter λ controls the balance:

- $\lambda = 0$: Pure safety optimization (may destroy image quality)
- $\lambda \approx 0.5$: Balanced tradeoff (recommended default)
- $\lambda > 1$: Prioritize quality preservation (may compromise safety)

6 Part V: Implementation Strategy

6.1 Algorithm: PPO (Proximal Policy Optimization)

We use PPO (Schulman et al., 2017) to train π_ϕ.

1. **Collect trajectories** $\tau = (z_T, a_T, \ldots, z_0)$ by running the steered ODE.
2. **Compute advantages** A_t based on the final Safety Reward.
3. **Update policy** π_ϕ using the clipped surrogate objective:

$$L^{\text{CLIP}}(\phi) = \mathbb{E}_t \left[\min \left(r_t(\phi) A_t, \text{clip}(r_t(\phi), 1 - \epsilon, 1 + \epsilon) A_t \right) \right]$$

where $r_t(\phi) = \frac{\pi_\phi(a_t|s_t)}{\pi_{\phi_{\text{old}}}(a_t|s_t)}$.

6.2 Three-Phase Deployment

Phase 1: The Probe (Sanity Check)

- Train a Linear Classifier on z_t to separate Safe/Unsafe concepts.
- **Purpose:** Verify that concepts are linearly separable (Abbasian et al. and Shen et al. suggest this is often true in latent spaces).
- Run Layer Sensitivity Analysis (Section 3) to identify optimal intervention layers.

Phase 2: The Training

- **Freeze Base Model** v_θ (following MIT 6.S184 best practices).
- **Train Policy** π_ϕ with PPO (Lamba et al. methodology).
- Use layer weights from sensitivity analysis in reward function (Equation 3).

Phase 3: Evaluation

- **SSR (Safety Success Rate):** Maximize. ↑
- **LPIPS:** Minimize perceptual distance to maintain fidelity. ↓
- **Transport Cost:** Minimize $\sum_t \|\Delta z_t\|^2$. ↓

7 References

1. **Lamba, P., Ravish, K., Kushwaha, A., & Kumar, P.** (2025). *Alignment and Safety of Diffusion Models via Reinforcement Learning and Reward Modeling: A Survey.* arXiv:2505.17352v1.
2. **Abbasian, M., Rajabzadeh, T., Moradipari, A., et al.** (2023). *Controlling the Latent Space of GANs through Reinforcement Learning: A Case Study on Task-based Image-to-Image Translation.* arXiv:2307.13978v1.
3. **Holderrieth, P. & Erives, E.** (2025). *MIT Class 6.S184: Generative AI With Stochastic Differential Equations.* https://diffusion.csail.mit.edu/.
4. **Song, Y., et al.** [30] (Cited in arXiv:2506.02070v2). Quote: "Creating data from noise is generative modeling."
5. **Wagenmaker, A., Nakamoto, M., Zhang, Y., et al.** (2025). *Steering Your Diffusion Policy with Latent Space Reinforcement Learning.* arXiv:2506.15799v2.
6. **Alain, G. & Bengio, Y.** (2016). *Understanding Intermediate Layers Using Linear Classifier Probes.* arXiv:1610.01644.
7. **Lu, S., Wang, Z., et al.** (2024). *MACE: Mass Concept Erasure in Diffusion Models.* CVPR 2024.
8. **Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O.** (2017). *Proximal Policy Optimization Algorithms.* arXiv:1707.06347.
9. **Shen, Y., Gu, J., Tang, X., & Zhou, B.** (2020). *Interpreting the Latent Space of GANs for Semantic Face Editing.* CVPR 2020. (InterFaceGAN)

A Appendix: Summary of Equations

Table 1: Key Equations in the Project Aether Framework

Component	Equation	Reference
ODE Dynamics	$\frac{dz}{dt} = v_\theta(z_t, t)$	MIT 6.S184
Steering	$\bar{z}_t = z_t + \pi_\phi(z_t, t)$	Wagenmaker et al.
Linear Probe	$\text{score}(z) = w \cdot z + b$	Alain & Bengio
Layer Sensitivity	$S_\ell = \text{Acc}_\ell \times \text{Qual}_\ell \times \text{Eff}_\ell$	This work
OT Reward	$J(\phi) = \mathbb{E}[R_{\text{safe}} - \lambda \sum_t \|a_t\|^2]$	Lamba et al.
Weighted Reward	$J(\phi) = \mathbb{E}[R_{\text{safe}} - \lambda \sum_t w_t \|a_t\|^2]$	This work

B Appendix: Notation Reference

Table 2: Notation Used Throughout This Document

Symbol	Meaning
z_t	Latent state at timestep t
v_θ	Base model velocity field (frozen)
π_ϕ	Steering policy (trained)
Δz_t	Steering vector at timestep t
$\bar{z}_t$	Steered latent state
w	Linear probe weight vector
b	Linear probe bias
S_ℓ	Layer sensitivity score
λ	Transport cost penalty coefficient
w_t	Layer weight (from sensitivity analysis)
R_{safe}	Safety reward ($+1$ or -1)
SSR	Safety Success Rate
LPIPS	Learned Perceptual Image Patch Similarity
FID	Fréchet Inception Distance
SSB	Semantic Single Boundary