GENERATIVE MODELING BY TRANSPORT: MATHEMATICAL FOUNDATIONS

Mathematical Foundations, Proofs, and Modern Unification

Ziseok Lee

April 30, 2026

The object: generative modeling as probability transport

Generative modeling is the problem of turning an easy law into a complex data law:

$$Z \sim \rho_0 \qquad \rightsquigarrow \qquad X \sim \rho_{\mathrm{data}}.$$

Modern methods differ mainly in how they realize a path of laws

$$(\rho_t)_{t\in[0,1]}, \qquad \rho_0 \to \rho_{\mathrm{data}}.$$

- Normalizing flows: invertible maps and change of variables.
- Continuous flows: ODEs and continuity equations.
- Diffusion models: SDEs, scores, and reverse-time dynamics.
- Flow matching / stochastic interpolants: finite-time bridges between laws.

WHY THIS IS MATHEMATICALLY INTERESTING

The same model can be viewed simultaneously as

$$\text{sampling algorithm} \quad + \quad \text{PDE for densities} \quad + \quad \text{stochastic process} \quad + \quad \text{variational principle.}$$

Understanding these equivalences is the main theme of the course.

The method: theorem-first, not recipe-first

We will not treat diffusion and flow models as engineering heuristics. For every major construction, we prove:

- existence of the probability path or process;
- the density evolution equation:

$$\partial_t \rho + \nabla \cdot (v\rho) = 0, \qquad \partial_t \rho = \mathcal{L}_t^* \rho;$$

- the reverse-time formula and score identity;
- the training objective from likelihood, ELBO, or projection;
- what approximation error controls: KL, likelihood, Wasserstein, or sampling error.

Concrete semester plan

- Weeks 1–2: mathematical preliminaries; measure, weak convergence, conditional expectation, Itô calculus, Fokker–Planck equations.
- Weeks 3–5: normalizing flows, CNFs, transport equations, likelihood identities.
- Weeks 6–8: score matching, denoising identities, DDPMs, SDEs, probability-flow ODEs.
- Weeks 9–11: flow matching, stochastic interpolants, Schrödinger bridges, optimal transport links.
- Weeks 12–14: discretization, convergence, likelihood control, proof-based student presentations.

Semester I builds the continuous Euclidean theory; **Semester II** abstracts and generalizes it.

Motivation: beyond Gaussian noise in $\mathbb{R}^d$

Semester I explains flows and diffusions in Euclidean space. Semester II asks which ideas survive when:

- the state space is discrete, categorical, manifold-valued, or multimodal;
- the process is not only an ODE or diffusion, but a general Markov process;
- generation uses jumps, CTMCs, hybrid dynamics, or latent spaces;
- training objectives are best understood on path space.

$$\text{Generative modeling} \quad \approx \quad \text{construct a Markov process with prescribed marginals.}$$

Unifying mathematical lens

The central object becomes the infinitesimal generator:

$$\mathcal{L}_t f(x) = \lim_{h \downarrow 0} \frac{\mathbb{E}[f(X_{t+h}) \mid X_t = x] - f(x)}{h}.$$

It unifies:

$$\begin{aligned} \text{ODEs} \quad &: \quad \mathcal{L}_t f = \nabla f \cdot v_t, \\ \text{Diffusions} \quad &: \quad \mathcal{L}_t f = b_t \cdot \nabla f + \tfrac{1}{2} a_t : \nabla^2 f, \\ \text{Jumps/CTMCs} \quad &: \quad \mathcal{L}_t f(x) = \textstyle\sum_y Q_t(y, x)\big(f(y) - f(x)\big). \end{aligned}$$

Concrete semester plan

- Weeks 1–2: Markov semigroups, generators, adjoints, Dynkin formula, martingale problems.
- Weeks 3–4: Kolmogorov forward/backward equations, time reversal, path measures, Girsanov-type identities.
- Weeks 5–6: discrete diffusion, CTMCs, masking processes, master equations, discrete scores.
- Weeks 7–8: generator matching, Bregman losses, jump processes, Markov superpositions.
- Weeks 9–10: general-state-space diffusion, manifolds, product spaces, multimodal generation.
- Weeks 11–12: latent diffusion, representation spaces, likelihood and ELBO in abstract state spaces.
- Weeks 13–14: current papers, open proof problems, final research presentations.

Learning goals

By the end, students should be able to:

- derive a generative model from a prescribed probability path;
- prove that its dynamics have the desired marginals;
- identify the correct training objective and its variational meaning;
- move between algorithms, PDEs, stochastic processes, and functional analysis;
- read modern generative-modeling papers at proof level, not only implementation level.

Pedagogical rule: every algorithm is introduced only after the theorem that justifies it.