

Stochastic Interpolants

A Unifying Framework for Flows and Diffusions (2025)

Ziyi Gao, Ruochuan Xu

June 2026

Stanford University

Outline

Motivation and framework

Structural results and implication for generative modeling

From Framework to Design Choices

Connections with other methods

Algorithmic aspects

Numerical results

Motivation and framework

From one distribution to another

Standard diffusion models learn the (reverse) dynamics from an easy-to-sample distribution (Gaussian) to a target distribution. The SDE is designable.

Interpolating two distributions

Given samples from a base density ρ_0 and a target density ρ_1 , build a bridge $\rho(t)$, $t \in [0, 1]$ from ρ_0 to ρ_1 .

Example process: for independent $x_0 \sim \rho_0$, $x_1 \sim \rho_1$, and $z \sim \mathcal{N}(0, Id)$,

$$x_t = (1 - t)x_0 + tx_1 + \sqrt{2t(1 - t)}z$$

- Such a process can bridge two arbitrary densities
- Since the process runs for finite-time, truncation error is avoided
- Can we learn the dynamics to build a generative model?

From one distribution to another

Standard diffusion models learn the (reverse) dynamics from an easy-to-sample distribution (Gaussian) to a target distribution. The SDE is designable.

Interpolating two distributions

Given samples from a base density ρ_0 and a target density ρ_1 , build a bridge $\rho(t)$, $t \in [0, 1]$ from ρ_0 to ρ_1 .

Example process: for independent $x_0 \sim \rho_0$, $x_1 \sim \rho_1$, and $z \sim \mathcal{N}(0, Id)$,

$$x_t = (1 - t)x_0 + tx_1 + \sqrt{2t(1 - t)}z$$

- Such a process can bridge two arbitrary densities
- Since the process runs for finite-time, truncation error is avoided
- Can we learn the dynamics to build a generative model?

From one distribution to another

Standard diffusion models learn the (reverse) dynamics from an easy-to-sample distribution (Gaussian) to a target distribution. The SDE is designable.

Interpolating two distributions

Given samples from a base density ρ_0 and a target density ρ_1 , build a bridge $\rho(t)$, $t \in [0, 1]$ from ρ_0 to ρ_1 .

Example process: for independent $x_0 \sim \rho_0$, $x_1 \sim \rho_1$, and $z \sim \mathcal{N}(0, Id)$,

$$x_t = (1 - t)x_0 + tx_1 + \sqrt{2t(1 - t)}z$$

- Such a process can bridge two arbitrary densities
- Since the process runs for finite-time, truncation error is avoided
- Can we learn the dynamics to build a generative model?

From one distribution to another

Standard diffusion models learn the (reverse) dynamics from an easy-to-sample distribution (Gaussian) to a target distribution. The SDE is designable.

Interpolating two distributions

Given samples from a base density ρ_0 and a target density ρ_1 , build a bridge $\rho(t)$, $t \in [0, 1]$ from ρ_0 to ρ_1 .

Example process: for independent $x_0 \sim \rho_0$, $x_1 \sim \rho_1$, and $z \sim \mathcal{N}(0, Id)$,

$$x_t = (1 - t)x_0 + tx_1 + \sqrt{2t(1 - t)}z$$

- Such a process can bridge two arbitrary densities
- Since the process runs for finite-time, truncation error is avoided
- Can we learn the dynamics to build a generative model?

From one distribution to another

Standard diffusion models learn the (reverse) dynamics from an easy-to-sample distribution (Gaussian) to a target distribution. The SDE is designable.

Interpolating two distributions

Given samples from a base density ρ_0 and a target density ρ_1 , build a bridge $\rho(t)$, $t \in [0, 1]$ from ρ_0 to ρ_1 .

Example process: for independent $x_0 \sim \rho_0$, $x_1 \sim \rho_1$, and $z \sim \mathcal{N}(0, Id)$,

$$x_t = (1 - t)x_0 + tx_1 + \sqrt{2t(1 - t)}z$$

- Such a process can bridge two arbitrary densities
- Since the process runs for finite-time, truncation error is avoided
- Can we learn the dynamics to build a generative model?

From one distribution to another

Standard diffusion models learn the (reverse) dynamics from an easy-to-sample distribution (Gaussian) to a target distribution. The SDE is designable.

Interpolating two distributions

Given samples from a base density ρ_0 and a target density ρ_1 , build a bridge $\rho(t)$, $t \in [0, 1]$ from ρ_0 to ρ_1 .

Example process: for independent $x_0 \sim \rho_0$, $x_1 \sim \rho_1$, and $z \sim \mathcal{N}(0, Id)$,

$$x_t = (1 - t)x_0 + tx_1 + \sqrt{2t(1 - t)}z$$

- Such a process can bridge two arbitrary densities
- Since the process runs for finite-time, truncation error is avoided
- Can we learn the dynamics to build a generative model?

Flexible design: many interpolants, same endpoints

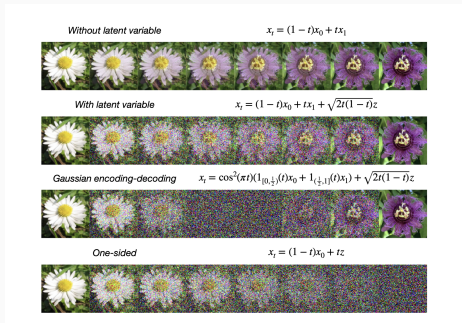


Figure 1: Different stochastic interpolants

Stochastic interpolant: a designable process

- The framework does not prescribe one canonical path from p_0 to p_1 .
- The latent variable $\gamma(t)z$ changes the intermediate laws $p(t)$; one-sided choices connect to standard diffusion (de)noising.

Flexible design: many interpolants, same endpoints

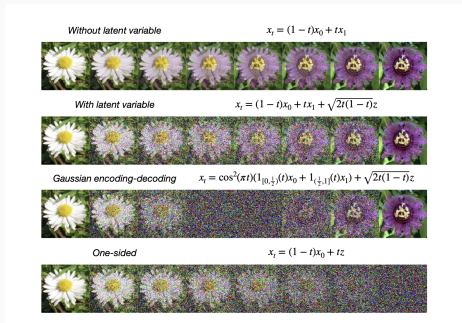


Figure 1: Different stochastic interpolants

Stochastic interpolant: a designable process

- The framework does not prescribe one canonical path from ρ_0 to ρ_1 .
- The latent variable $\gamma(t)z$ changes the intermediate laws $\rho(t)$; one-sided choices connect to standard diffusion (de)noising.

Flexible design: many interpolants, same endpoints

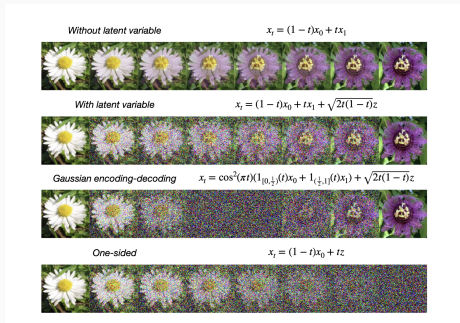


Figure 1: Different stochastic interpolants

Stochastic interpolant: a designable process

- The framework does not prescribe one canonical path from ρ_0 to ρ_1 .
- The latent variable $\gamma(t)z$ changes the intermediate laws $\rho(t)$; one-sided choices connect to standard diffusion (de)noising.

Main contribution

Building a generative model with a stochastic interpolant:



Key observation

The paper separates two design choices:

which density path $\rho(t)$ to use vs. how to sample that path.

Main contribution

Building a generative model with a stochastic interpolant:



Key observation

The paper separates two design choices:

which density path $\rho(t)$ to use vs. how to sample that path.

Main contribution

Building a generative model with a stochastic interpolant:



Key observation

The paper separates two design choices:

which density path $\rho(t)$ to use vs. how to sample that path.

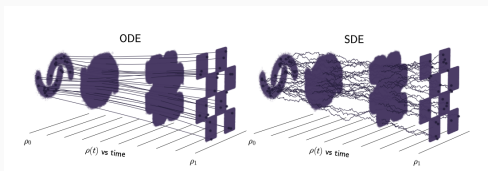


Figure 2: Different stochastic processes can have the same marginals

Structural results and implication for generative modeling

Definition: stochastic interpolant

Sample-level construction

Let $(x_0, x_1) \sim \nu$ where ν has marginals ρ_0 and ρ_1 , and let $z \sim \mathcal{N}(0, Id)$ be independent. Define

$$x_t = I(t, x_0, x_1) + \gamma(t)z, \quad t \in [0, 1].$$

Endpoint conditions

$$I(0, x_0, x_1) = x_0, \quad I(1, x_0, x_1) = x_1,$$

$$\gamma(0) = \gamma(1) = 0.$$

So $x_0 \sim \rho_0$ and $x_1 \sim \rho_1$ exactly.

Design components

- interpolation map I ;
- coupling ν between endpoints;
- latent noise schedule $\gamma(t)$.

For sampling, we mostly care about marginal laws of the stochastic interpolant process.

Question

What PDE is satisfied by the law $\rho(t) = \mathcal{L}(x_t)$?

Definition: stochastic interpolant

Sample-level construction

Let $(x_0, x_1) \sim \nu$ where ν has marginals ρ_0 and ρ_1 , and let $z \sim \mathcal{N}(0, Id)$ be independent. Define

$$x_t = I(t, x_0, x_1) + \gamma(t)z, \quad t \in [0, 1].$$

Endpoint conditions

$$I(0, x_0, x_1) = x_0, \quad I(1, x_0, x_1) = x_1,$$

$$\gamma(0) = \gamma(1) = 0.$$

So $x_0 \sim \rho_0$ and $x_1 \sim \rho_1$ exactly.

Design components

- interpolation map I ;
- coupling ν between endpoints;
- latent noise schedule $\gamma(t)$.

For sampling, we mostly care about marginal laws of the stochastic interpolant process.

Question

What PDE is satisfied by the law $\rho(t) = \mathcal{L}(x_t)$?

Definition: stochastic interpolant

Sample-level construction

Let $(x_0, x_1) \sim \nu$ where ν has marginals ρ_0 and ρ_1 , and let $z \sim \mathcal{N}(0, Id)$ be independent. Define

$$x_t = I(t, x_0, x_1) + \gamma(t)z, \quad t \in [0, 1].$$

Endpoint conditions

$$I(0, x_0, x_1) = x_0, \quad I(1, x_0, x_1) = x_1,$$

$$\gamma(0) = \gamma(1) = 0.$$

So $x_0 \sim \rho_0$ and $x_1 \sim \rho_1$ exactly.

Design components

- interpolation map I ;
- coupling ν between endpoints;
- latent noise schedule $\gamma(t)$.

For sampling, we mostly care about marginal laws of the stochastic interpolant process.

Question

What PDE is satisfied by the law $\rho(t) = \mathcal{L}(x_t)$?

Definition: stochastic interpolant

Sample-level construction

Let $(x_0, x_1) \sim \nu$ where ν has marginals ρ_0 and ρ_1 , and let $z \sim \mathcal{N}(0, Id)$ be independent. Define

$$x_t = I(t, x_0, x_1) + \gamma(t)z, \quad t \in [0, 1].$$

Endpoint conditions

$$I(0, x_0, x_1) = x_0, \quad I(1, x_0, x_1) = x_1,$$

$$\gamma(0) = \gamma(1) = 0.$$

So $x_0 \sim \rho_0$ and $x_1 \sim \rho_1$ exactly.

Design components

- interpolation map I ;
- coupling ν between endpoints;
- latent noise schedule $\gamma(t)$.

For sampling, we mostly care about marginal laws of the stochastic interpolant process.

Question

What PDE is satisfied by the law $\rho(t) = \mathcal{L}(x_t)$?

Definition: stochastic interpolant

Sample-level construction

Let $(x_0, x_1) \sim \nu$ where ν has marginals ρ_0 and ρ_1 , and let $z \sim \mathcal{N}(0, Id)$ be independent. Define

$$x_t = I(t, x_0, x_1) + \gamma(t)z, \quad t \in [0, 1].$$

Endpoint conditions

$$I(0, x_0, x_1) = x_0, \quad I(1, x_0, x_1) = x_1,$$

$$\gamma(0) = \gamma(1) = 0.$$

So $x_0 \sim \rho_0$ and $x_1 \sim \rho_1$ exactly.

Design components

- interpolation map I ;
- coupling ν between endpoints;
- latent noise schedule $\gamma(t)$.

For sampling, we mostly care about marginal laws of the stochastic interpolant process.

Question

What PDE is satisfied by the law $\rho(t) = \mathcal{L}(x_t)$?

Main structural result: a transport equation

Theorem 6, informal statement

The law of x_t has a density $\rho(t, x)$ satisfying

$$\partial_t \rho(t, x) + \nabla \cdot (b(t, x) \rho(t, x)) = 0, \quad \rho(0) = \rho_0, \quad \rho(1) = \rho_1.$$

The velocity field is the conditional average instantaneous motion:

$$b(t, x) = \mathbb{E}[\dot{x}_t \mid x_t = x] = \mathbb{E}[\partial_t I(t, x_0, x_1) + \dot{\gamma}(t)z \mid x_t = x].$$

Interpretation: the random bridge defines a deterministic density evolution.

The drift b is not chosen independently; it is induced by the interpolant.

Main structural result: a transport equation

Theorem 6, informal statement

The law of x_t has a density $\rho(t, x)$ satisfying

$$\partial_t \rho(t, x) + \nabla \cdot (b(t, x) \rho(t, x)) = 0, \quad \rho(0) = \rho_0, \quad \rho(1) = \rho_1.$$

The velocity field is the conditional average instantaneous motion:

$$b(t, x) = \mathbb{E}[\dot{x}_t \mid x_t = x] = \mathbb{E}[\partial_t I(t, x_0, x_1) + \dot{\gamma}(t)z \mid x_t = x].$$

Interpretation: the random bridge defines a deterministic density evolution.

The drift b is not chosen independently; it is induced by the interpolant.

Learnability: fields are regression targets

Velocity regression

The velocity b is the minimizer of the quadratic objective

$$\mathcal{L}_b[\hat{b}] = \int_0^1 \mathbb{E} \left[\frac{1}{2} |\hat{b}(t, x_t)|^2 - (\partial_t I(t, x_0, x_1) + \dot{\gamma}(t)z) \cdot \hat{b}(t, x_t) \right] dt.$$

Score and denoiser regression

The score of the intermediate density can be written as

$$s(t, x) = \nabla \log \rho(t, x) = -\gamma(t)^{-1} \eta_z(t, x), \quad \eta_z(t, x) = \mathbb{E}[z \mid x_t = x].$$

Thus, one can learn either the score s or the denoiser η_z through a quadratic loss $\mathcal{L}_s[\hat{s}]$.

Message for generative modeling

Training does not require solving the ODE/SDE. It reduces to supervised regression on samples of (x_0, x_1, z, t) .

Learnability: fields are regression targets

Velocity regression

The velocity b is the minimizer of the quadratic objective

$$\mathcal{L}_b[\hat{b}] = \int_0^1 \mathbb{E} \left[\frac{1}{2} |\hat{b}(t, x_t)|^2 - (\partial_t I(t, x_0, x_1) + \dot{\gamma}(t)z) \cdot \hat{b}(t, x_t) \right] dt.$$

Score and denoiser regression

The score of the intermediate density can be written as

$$s(t, x) = \nabla \log \rho(t, x) = -\gamma(t)^{-1} \eta_z(t, x), \quad \eta_z(t, x) = \mathbb{E}[z \mid x_t = x].$$

Thus, one can learn either the score s or the denoiser η_z through a quadratic loss $\mathcal{L}_s[\hat{s}]$.

Message for generative modeling

Training does not require solving the ODE/SDE. It reduces to supervised regression on samples of (x_0, x_1, z, t) .

Learnability: fields are regression targets

Velocity regression

The velocity b is the minimizer of the quadratic objective

$$\mathcal{L}_b[\hat{b}] = \int_0^1 \mathbb{E} \left[\frac{1}{2} |\hat{b}(t, x_t)|^2 - (\partial_t I(t, x_0, x_1) + \dot{\gamma}(t)z) \cdot \hat{b}(t, x_t) \right] dt.$$

Score and denoiser regression

The score of the intermediate density can be written as

$$s(t, x) = \nabla \log \rho(t, x) = -\gamma(t)^{-1} \eta_z(t, x), \quad \eta_z(t, x) = \mathbb{E}[z \mid x_t = x].$$

Thus, one can learn either the score s or the denoiser η_z through a quadratic loss $\mathcal{L}_s[\hat{s}]$.

Message for generative modeling

Training does not require solving the ODE/SDE. It reduces to supervised regression on samples of (x_0, x_1, z, t) .

From the same density evolution to an ODE or SDE

The density evolution $\rho(t)$ of a stochastic interpolant can be realized by a family of stochastic processes.

Probability-flow ODE

ρ solves

$$\partial_t \rho + \nabla \cdot (b\rho) = 0,$$

which is the transport equation for

$$\frac{dX_t}{dt} = b(t, X_t), \quad X_0 \sim \rho_0.$$

Forward Fokker-Planck / SDE

For any $\epsilon(t) \geq 0$, ρ solves

$$\partial_t \rho + \nabla \cdot (b_F \rho) = \epsilon(t) \Delta \rho, \quad b_F = b + \epsilon s,$$

which is the forward FP equation for

$$dX_t^F = b_F(t, X_t^F) dt + \sqrt{2\epsilon(t)} dW_t.$$

Equivalence of marginals

With the true b and s , both samplers have the same one-time marginals:

$$X_t \sim X_t^F \sim x_t \sim \rho(t), \quad t \in [0, 1].$$

A backward Fokker-Planck/SDE counterpart can also be formulated.

From the same density evolution to an ODE or SDE

The density evolution $\rho(t)$ of a stochastic interpolant can be realized by a family of stochastic processes.

Probability-flow ODE

ρ solves

$$\partial_t \rho + \nabla \cdot (b\rho) = 0,$$

which is the transport equation for

$$\frac{dX_t}{dt} = b(t, X_t), \quad X_0 \sim \rho_0.$$

Forward Fokker-Planck / SDE

For any $\epsilon(t) \geq 0$, ρ solves

$$\partial_t \rho + \nabla \cdot (b_F \rho) = \epsilon(t) \Delta \rho, \quad b_F = b + \epsilon s,$$

which is the forward FP equation for

$$dX_t^F = b_F(t, X_t^F) dt + \sqrt{2\epsilon(t)} dW_t.$$

Equivalence of marginals

With the true b and s , both samplers have the same one-time marginals:

$$X_t \sim X_t^F \sim x_t \sim \rho(t), \quad t \in [0, 1].$$

A backward Fokker-Planck/SDE counterpart can also be formulated.

From the same density evolution to an ODE or SDE

The density evolution $\rho(t)$ of a stochastic interpolant can be realized by a family of stochastic processes.

Probability-flow ODE

ρ solves

$$\partial_t \rho + \nabla \cdot (b\rho) = 0,$$

which is the transport equation for

$$\frac{dX_t}{dt} = b(t, X_t), \quad X_0 \sim \rho_0.$$

Forward Fokker-Planck / SDE

For any $\epsilon(t) \geq 0$, ρ solves

$$\partial_t \rho + \nabla \cdot (b_F \rho) = \epsilon(t) \Delta \rho, \quad b_F = b + \epsilon s,$$

which is the forward FP equation for

$$dX_t^F = b_F(t, X_t^F) dt + \sqrt{2\epsilon(t)} dW_t.$$

Equivalence of marginals

With the true b and s , both samplers have the same one-time marginals:

$$X_t \sim X_t^F \sim x_t \sim \rho(t), \quad t \in [0, 1].$$

A backward Fokker-Planck/SDE counterpart can also be formulated.

From the same density evolution to an ODE or SDE

The density evolution $\rho(t)$ of a stochastic interpolant can be realized by a family of stochastic processes.

Probability-flow ODE

ρ solves

$$\partial_t \rho + \nabla \cdot (b\rho) = 0,$$

which is the transport equation for

$$\frac{dX_t}{dt} = b(t, X_t), \quad X_0 \sim \rho_0.$$

Forward Fokker–Planck / SDE

For any $\epsilon(t) \geq 0$, ρ solves

$$\partial_t \rho + \nabla \cdot (b_F \rho) = \epsilon(t) \Delta \rho, \quad b_F = b + \epsilon s,$$

which is the forward FP equation for

$$dX_t^F = b_F(t, X_t^F) dt + \sqrt{2\epsilon(t)} dW_t.$$

Equivalence of marginals

With the true b and s , both samplers have the same one-time marginals:

$$X_t \sim X_t^F \sim x_t \sim \rho(t), \quad t \in [0, 1].$$

A backward Fokker–Planck/SDE counterpart can also be formulated.

Likelihood control: why ODE and SDE differ

Since the drift b and the score s are imperfectly estimated in practice, we wish to understand the stability of sampling accuracy with respect to the estimation errors.

ODE / transport

$$\partial_t \rho + \nabla \cdot (b \rho) = 0,$$

$$\partial_t \hat{\rho} + \nabla \cdot (\hat{b} \hat{\rho}) = 0.$$

$$\text{KL}(\rho(1) \parallel \hat{\rho}(1)) = \int_0^1 \int (\nabla \log \hat{\rho} - \nabla \log \rho) \cdot (\hat{b} - b) \rho \, dx \, dt.$$

SDE / Fokker-Planck

$$\partial_t \rho + \nabla \cdot (b_F \rho) = \epsilon \Delta \rho,$$

$$\partial_t \hat{\rho} + \nabla \cdot (\hat{b}_F \hat{\rho}) = \epsilon \Delta \hat{\rho}.$$

$$\text{KL}(\rho(1) \parallel \hat{\rho}(1)) \leq \frac{1}{4\epsilon} \int_0^1 \int |\hat{b}_F - b_F|^2 \rho \, dx \, dt.$$

Consequence

- For ODEs, drift error is multiplied by the unknown score mismatch $\nabla \log \hat{\rho} - \nabla \log \rho$.
- For SDEs, the diffusion term contributes a negative Fisher-divergence term, yielding control by the SDE drift error $|\hat{b}_F - b_F|^2$.

Likelihood control: why ODE and SDE differ

Since the drift b and the score s are imperfectly estimated in practice, we wish to understand the stability of sampling accuracy with respect to the estimation errors.

ODE / transport

$$\partial_t \rho + \nabla \cdot (b \rho) = 0,$$

$$\partial_t \hat{\rho} + \nabla \cdot (\hat{b} \hat{\rho}) = 0.$$

$$\text{KL}(\rho(1) \parallel \hat{\rho}(1)) = \int_0^1 \int (\nabla \log \hat{\rho} - \nabla \log \rho) \cdot (\hat{b} - b) \rho \, dx \, dt.$$

SDE / Fokker-Planck

$$\partial_t \rho + \nabla \cdot (b_F \rho) = \epsilon \Delta \rho,$$

$$\partial_t \hat{\rho} + \nabla \cdot (\hat{b}_F \hat{\rho}) = \epsilon \Delta \hat{\rho}.$$

$$\text{KL}(\rho(1) \parallel \hat{\rho}(1)) \leq \frac{1}{4\epsilon} \int_0^1 \int |\hat{b}_F - b_F|^2 \rho \, dx \, dt.$$

Consequence

- For ODEs, drift error is multiplied by the unknown score mismatch $\nabla \log \hat{\rho} - \nabla \log \rho$.
- For SDEs, the diffusion term contributes a negative Fisher-divergence term, yielding control by the SDE drift error $|\hat{b}_F - b_F|^2$.

Likelihood control: why ODE and SDE differ

Since the drift b and the score s are imperfectly estimated in practice, we wish to understand the stability of sampling accuracy with respect to the estimation errors.

ODE / transport

$$\partial_t \rho + \nabla \cdot (b \rho) = 0,$$

$$\partial_t \hat{\rho} + \nabla \cdot (\hat{b} \hat{\rho}) = 0.$$

$$\text{KL}(\rho(1) \parallel \hat{\rho}(1)) = \int_0^1 \int (\nabla \log \hat{\rho} - \nabla \log \rho) \cdot (\hat{b} - b) \rho \, dx \, dt.$$

SDE / Fokker–Planck

$$\partial_t \rho + \nabla \cdot (b_F \rho) = \epsilon \Delta \rho,$$

$$\partial_t \hat{\rho} + \nabla \cdot (\hat{b}_F \hat{\rho}) = \epsilon \Delta \hat{\rho}.$$

$$\text{KL}(\rho(1) \parallel \hat{\rho}(1)) \leq \frac{1}{4\epsilon} \int_0^1 \int |\hat{b}_F - b_F|^2 \rho \, dx \, dt.$$

Consequence

- For ODEs, drift error is multiplied by the unknown score mismatch $\nabla \log \hat{\rho} - \nabla \log \rho$.
- For SDEs, the diffusion term contributes a negative Fisher-divergence term, yielding control by the SDE drift error $|\hat{b}_F - b_F|^2$.

Likelihood control: why ODE and SDE differ

Since the drift b and the score s are imperfectly estimated in practice, we wish to understand the stability of sampling accuracy with respect to the estimation errors.

ODE / transport

$$\partial_t \rho + \nabla \cdot (b \rho) = 0,$$

$$\partial_t \hat{\rho} + \nabla \cdot (\hat{b} \hat{\rho}) = 0.$$

$$\text{KL}(\rho(1) \parallel \hat{\rho}(1)) = \int_0^1 \int (\nabla \log \hat{\rho} - \nabla \log \rho) \cdot (\hat{b} - b) \rho \, dx \, dt.$$

SDE / Fokker–Planck

$$\partial_t \rho + \nabla \cdot (b_F \rho) = \epsilon \Delta \rho,$$

$$\partial_t \hat{\rho} + \nabla \cdot (\hat{b}_F \hat{\rho}) = \epsilon \Delta \hat{\rho}.$$

$$\text{KL}(\rho(1) \parallel \hat{\rho}(1)) \leq \frac{1}{4\epsilon} \int_0^1 \int |\hat{b}_F - b_F|^2 \rho \, dx \, dt.$$

Consequence

- For ODEs, drift error is multiplied by the unknown score mismatch $\nabla \log \hat{\rho} - \nabla \log \rho$.
- For SDEs, the diffusion term contributes a negative Fisher-divergence term, yielding control by the SDE drift error $|\hat{b}_F - b_F|^2$.

Likelihood control: why ODE and SDE differ

Since the drift b and the score s are imperfectly estimated in practice, we wish to understand the stability of sampling accuracy with respect to the estimation errors.

ODE / transport

$$\partial_t \rho + \nabla \cdot (b \rho) = 0,$$

$$\partial_t \hat{\rho} + \nabla \cdot (\hat{b} \hat{\rho}) = 0.$$

$$\text{KL}(\rho(1) \parallel \hat{\rho}(1)) = \int_0^1 \int (\nabla \log \hat{\rho} - \nabla \log \rho) \cdot (\hat{b} - b) \rho \, dx \, dt.$$

SDE / Fokker–Planck

$$\partial_t \rho + \nabla \cdot (b_F \rho) = \epsilon \Delta \rho,$$

$$\partial_t \hat{\rho} + \nabla \cdot (\hat{b}_F \hat{\rho}) = \epsilon \Delta \hat{\rho}.$$

$$\text{KL}(\rho(1) \parallel \hat{\rho}(1)) \leq \frac{1}{4\epsilon} \int_0^1 \int |\hat{b}_F - b_F|^2 \rho \, dx \, dt.$$

Consequence

- For ODEs, drift error is multiplied by the unknown score mismatch $\nabla \log \hat{\rho} - \nabla \log \rho$.
- For SDEs, the diffusion term contributes a negative Fisher-divergence term, yielding control by the SDE drift error $|\hat{b}_F - b_F|^2$.

Theorem 23: It suffices to learn b and s well

For the learned forward SDE drift

$$\hat{b}_F(t, x) = \hat{b}(t, x) + \epsilon \hat{s}(t, x),$$

Let $\hat{\rho}$ be the solution to the FP equation with drift $\hat{b}_F$ and diffusion coefficient ϵId . Theorem 23 gives the likelihood control

$$\text{KL}(\rho_1 \parallel \hat{\rho}(1)) \leq \frac{1}{2\epsilon} \left(\mathcal{L}_b[\hat{b}] - \min_{\tilde{b}} \mathcal{L}_b[\tilde{b}] \right) + \frac{\epsilon}{2} \left(\mathcal{L}_s[\hat{s}] - \min_{\tilde{s}} \mathcal{L}_s[\tilde{s}] \right).$$

Takeaway

- With exact fields, ODE and SDE samplers share the same marginals $\rho(t)$.
- With learned fields, SDE sampling has a direct KL guarantee from velocity and score regression errors.
- ODE sampling with learned $\hat{b}$ lacks this guarantee unless additional score/Fisher control is available.

Theorem 23: It suffices to learn b and s well

For the learned forward SDE drift

$$\hat{b}_F(t, x) = \hat{b}(t, x) + \epsilon \hat{s}(t, x),$$

Let $\hat{\rho}$ be the solution to the FP equation with drift $\hat{b}_F$ and diffusion coefficient ϵId . Theorem 23 gives the likelihood control

$$\text{KL}(\rho_1 \parallel \hat{\rho}(1)) \leq \frac{1}{2\epsilon} \left(\mathcal{L}_b[\hat{b}] - \min_{\tilde{b}} \mathcal{L}_b[\tilde{b}] \right) + \frac{\epsilon}{2} \left(\mathcal{L}_s[\hat{s}] - \min_{\tilde{s}} \mathcal{L}_s[\tilde{s}] \right).$$

Takeaway

- With exact fields, ODE and SDE samplers share the same marginals $\rho(t)$.
- With learned fields, SDE sampling has a direct KL guarantee from velocity and score regression errors.
- ODE sampling with learned $\hat{b}$ lacks this guarantee unless additional score/Fisher control is available.

Theorem 23: It suffices to learn b and s well

For the learned forward SDE drift

$$\hat{b}_F(t, x) = \hat{b}(t, x) + \epsilon \hat{s}(t, x),$$

Let $\hat{\rho}$ be the solution to the FP equation with drift $\hat{b}_F$ and diffusion coefficient ϵId . Theorem 23 gives the likelihood control

$$\text{KL}(\rho_1 \parallel \hat{\rho}(1)) \leq \frac{1}{2\epsilon} \left(\mathcal{L}_b[\hat{b}] - \min_{\tilde{b}} \mathcal{L}_b[\tilde{b}] \right) + \frac{\epsilon}{2} \left(\mathcal{L}_s[\hat{s}] - \min_{\tilde{s}} \mathcal{L}_s[\tilde{s}] \right).$$

Takeaway

- With exact fields, ODE and SDE samplers share the same marginals $\rho(t)$.
- With learned fields, SDE sampling has a direct KL guarantee from velocity and score regression errors.
- ODE sampling with learned $\hat{b}$ lacks this guarantee unless additional score/Fisher control is available.

Theorem 23: It suffices to learn b and s well

For the learned forward SDE drift

$$\hat{b}_F(t, x) = \hat{b}(t, x) + \epsilon \hat{s}(t, x),$$

Let $\hat{\rho}$ be the solution to the FP equation with drift $\hat{b}_F$ and diffusion coefficient ϵId . Theorem 23 gives the likelihood control

$$\text{KL}(\rho_1 \parallel \hat{\rho}(1)) \leq \frac{1}{2\epsilon} \left(\mathcal{L}_b[\hat{b}] - \min_{\tilde{b}} \mathcal{L}_b[\tilde{b}] \right) + \frac{\epsilon}{2} \left(\mathcal{L}_s[\hat{s}] - \min_{\tilde{s}} \mathcal{L}_s[\tilde{s}] \right).$$

Takeaway

- With exact fields, ODE and SDE samplers share the same marginals $\rho(t)$.
- With learned fields, SDE sampling has a direct KL guarantee from velocity and score regression errors.
- ODE sampling with learned $\hat{b}$ lacks this guarantee unless additional score/Fisher control is available.

From Framework to Design Choices

Design choices: what is not fixed by the endpoints?

Stochastic interpolant

Given endpoint samples $x_0 \sim \rho_0$ and $x_1 \sim \rho_1$, the framework builds

$$x_t = I(t, x_0, x_1) + \gamma(t)z, \quad z \sim \mathcal{N}(0, I_d), \quad t \in [0, 1].$$

The endpoint constraints only require

$$x_{t=0} \sim \rho_0, \quad x_{t=1} \sim \rho_1.$$

$I(t, x_0, x_1)$

Interpolation map

Controls how individual
endpoint samples are
combined.

$\nu(dx_0, dx_1)$

Endpoint coupling

Controls which x_0 samples are
paired with which x_1 samples.

$\gamma(t)$

Latent noise schedule

Controls how much noise is
injected along the bridge.

Main message

The endpoints ρ_0 and ρ_1 do not determine a unique path. Different choices of I , ν , and $\gamma(t)$ produce different intermediate densities $\rho(t)$, and therefore different learned generative models.

Design choices: what is not fixed by the endpoints?

Stochastic interpolant

Given endpoint samples $x_0 \sim \rho_0$ and $x_1 \sim \rho_1$, the framework builds

$$x_t = I(t, x_0, x_1) + \gamma(t)z, \quad z \sim \mathcal{N}(0, I_d), \quad t \in [0, 1].$$

The endpoint constraints only require

$$x_{t=0} \sim \rho_0, \quad x_{t=1} \sim \rho_1.$$

$I(t, x_0, x_1)$

Interpolation map

Controls how individual
endpoint samples are
combined.

$\nu(dx_0, dx_1)$

Endpoint coupling

Controls which x_0 samples are
paired with which x_1 samples.

$\gamma(t)$

Latent noise schedule

Controls how much noise is
injected along the bridge.

Main message

The endpoints ρ_0 and ρ_1 do not determine a unique path. Different choices of I , ν , and $\gamma(t)$ produce different intermediate densities $\rho(t)$, and therefore different learned generative models.

Design choices: what is not fixed by the endpoints?

Stochastic interpolant

Given endpoint samples $x_0 \sim \rho_0$ and $x_1 \sim \rho_1$, the framework builds

$$x_t = I(t, x_0, x_1) + \gamma(t)z, \quad z \sim \mathcal{N}(0, I_d), \quad t \in [0, 1].$$

The endpoint constraints only require

$$x_{t=0} \sim \rho_0, \quad x_{t=1} \sim \rho_1.$$

$I(t, x_0, x_1)$

Interpolation map

Controls how individual
endpoint samples are
combined.

$\nu(dx_0, dx_1)$

Endpoint coupling

Controls which x_0 samples are
paired with which x_1 samples.

$\gamma(t)$

Latent noise schedule

Controls how much noise is
injected along the bridge.

Main message

The endpoints ρ_0 and ρ_1 do not determine a unique path. Different choices of I , ν , and $\gamma(t)$ produce different intermediate densities $\rho(t)$, and therefore different learned generative models.

Spatially linear interpolants

A simple but flexible design class

A common choice is to make the interpolant linear in the endpoint samples:

$$x_t^{\text{lin}} = \alpha(t)x_0 + \beta(t)x_1 + \gamma(t)z, \quad z \sim \mathcal{N}(0, I_d).$$

The endpoint conditions are

$$\alpha(0) = 1, \quad \alpha(1) = 0, \quad \beta(0) = 0, \quad \beta(1) = 1, \quad \gamma(0) = \gamma(1) = 0.$$

For this class, the relevant conditional expectations are

$$\eta_0(t, x) = \mathbb{E}[x_0 \mid x_t = x], \quad \eta_1(t, x) = \mathbb{E}[x_1 \mid x_t = x], \quad \eta_z(t, x) = \mathbb{E}[z \mid x_t = x].$$

Factorized velocity and score

$$b(t, x) = \dot{\alpha}(t)\eta_0(t, x) + \dot{\beta}(t)\eta_1(t, x) + \dot{\gamma}(t)\eta_z(t, x), \quad s(t, x) = -\gamma(t)^{-1}\eta_z(t, x).$$

Main message

The design functions $\alpha(t), \beta(t), \gamma(t)$ directly shape the intermediate density path and the fields that the model has to learn.

Spatially linear interpolants

A simple but flexible design class

A common choice is to make the interpolant linear in the endpoint samples:

$$x_t^{\text{lin}} = \alpha(t)x_0 + \beta(t)x_1 + \gamma(t)z, \quad z \sim \mathcal{N}(0, I_d).$$

The endpoint conditions are

$$\alpha(0) = 1, \quad \alpha(1) = 0, \quad \beta(0) = 0, \quad \beta(1) = 1, \quad \gamma(0) = \gamma(1) = 0.$$

For this class, the relevant conditional expectations are

$$\eta_0(t, x) = \mathbb{E}[x_0 \mid x_t = x], \quad \eta_1(t, x) = \mathbb{E}[x_1 \mid x_t = x], \quad \eta_z(t, x) = \mathbb{E}[z \mid x_t = x].$$

Factorized velocity and score

$$b(t, x) = \dot{\alpha}(t)\eta_0(t, x) + \dot{\beta}(t)\eta_1(t, x) + \dot{\gamma}(t)\eta_z(t, x), \quad s(t, x) = -\gamma(t)^{-1}\eta_z(t, x).$$

Main message

The design functions $\alpha(t), \beta(t), \gamma(t)$ directly shape the intermediate density path and the fields that the model has to learn.

Spatially linear interpolants

A simple but flexible design class

A common choice is to make the interpolant linear in the endpoint samples:

$$x_t^{\text{lin}} = \alpha(t)x_0 + \beta(t)x_1 + \gamma(t)z, \quad z \sim \mathcal{N}(0, I_d).$$

The endpoint conditions are

$$\alpha(0) = 1, \quad \alpha(1) = 0, \quad \beta(0) = 0, \quad \beta(1) = 1, \quad \gamma(0) = \gamma(1) = 0.$$

For this class, the relevant conditional expectations are

$$\eta_0(t, x) = \mathbb{E}[x_0 \mid x_t = x], \quad \eta_1(t, x) = \mathbb{E}[x_1 \mid x_t = x], \quad \eta_z(t, x) = \mathbb{E}[z \mid x_t = x].$$

Factorized velocity and score

$$b(t, x) = \dot{\alpha}(t)\eta_0(t, x) + \dot{\beta}(t)\eta_1(t, x) + \dot{\gamma}(t)\eta_z(t, x), \quad s(t, x) = -\gamma(t)^{-1}\eta_z(t, x).$$

Main message

The design functions $\alpha(t), \beta(t), \gamma(t)$ directly shape the intermediate density path and the fields that the model has to learn.

Spatially linear interpolants

A simple but flexible design class

A common choice is to make the interpolant linear in the endpoint samples:

$$x_t^{\text{lin}} = \alpha(t)x_0 + \beta(t)x_1 + \gamma(t)z, \quad z \sim \mathcal{N}(0, I_d).$$

The endpoint conditions are

$$\alpha(0) = 1, \quad \alpha(1) = 0, \quad \beta(0) = 0, \quad \beta(1) = 1, \quad \gamma(0) = \gamma(1) = 0.$$

For this class, the relevant conditional expectations are

$$\eta_0(t, x) = \mathbb{E}[x_0 \mid x_t = x], \quad \eta_1(t, x) = \mathbb{E}[x_1 \mid x_t = x], \quad \eta_z(t, x) = \mathbb{E}[z \mid x_t = x].$$

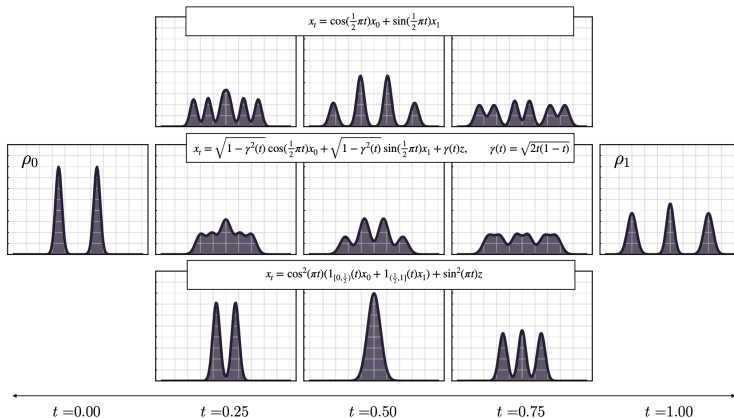
Factorized velocity and score

$$b(t, x) = \dot{\alpha}(t)\eta_0(t, x) + \dot{\beta}(t)\eta_1(t, x) + \dot{\gamma}(t)\eta_z(t, x), \quad s(t, x) = -\gamma(t)^{-1}\eta_z(t, x).$$

Main message

The design functions $\alpha(t), \beta(t), \gamma(t)$ directly shape the intermediate density path and the fields that the model has to learn.

Examples: different interpolants, same endpoints



Connections with other methods

Connection to score-based diffusion

OU-based score diffusion

$$dZ_\tau = -Z_\tau d\tau + \sqrt{2} dW_\tau, \quad Z_0 = x_1 \sim \rho_1.$$

The solution satisfies

$$Z_\tau \stackrel{d}{=} e^{-\tau} x_1 + \sqrt{1 - e^{-2\tau}} z, \quad z \sim \mathcal{N}(0, I_d).$$

Time reparameterization

Let

$$t = e^{-\tau}.$$

Then

$$Z_\tau \stackrel{d}{=} tx_1 + \sqrt{1 - t^2} z.$$

Diffusion as a one-sided interpolant

The reparameterized OU process becomes

$$x_t^{\text{os,lin}} = \alpha(t)z + \beta(t)x_1, \quad \alpha(t) = \sqrt{1-t^2}, \quad \beta(t) = t.$$

Interpretation

This is exactly a spatially linear one-sided stochastic interpolant.

Takeaway

Score-based diffusion and stochastic localization are not separate from the stochastic interpolant framework. They are recovered by a special choice of one-sided interpolant and the time change

$$\tau = -\log t.$$

Algorithmic aspects

Algorithmic view: how to train and sample

Step 1: Generate supervised training pairs

Sample

$$x_0 \sim \rho_0, \quad x_1 \sim \rho_1, \quad z \sim \mathcal{N}(0, I_d), \quad t \sim \text{Unif}([0, 1]).$$

Then construct an interpolant sample

$$x_t = I(t, x_0, x_1) + \gamma(t)z.$$

Step 2: Learn the fields by regression

The model learns conditional fields from samples of (t, x_t) :

$$\hat{b}(t, x) \approx \mathbb{E}[\dot{x}_t \mid x_t = x], \quad \hat{\eta}_z(t, x) \approx \mathbb{E}[z \mid x_t = x].$$

Equivalently,

$$\hat{s}(t, x) \approx -\gamma(t)^{-1} \hat{\eta}_z(t, x).$$

Step 3: Sampling is chosen after training

Once $\hat{b}$ and $\hat{s}$ are learned, samples from ρ_1 can be generated by integrating an ODE or SDE from the base distribution ρ_0 .

$$\text{train by regression} \implies \text{sample by ODE or SDE.}$$

Algorithmic view: how to train and sample

Step 1: Generate supervised training pairs

Sample

$$x_0 \sim \rho_0, \quad x_1 \sim \rho_1, \quad z \sim \mathcal{N}(0, I_d), \quad t \sim \text{Unif}([0, 1]).$$

Then construct an interpolant sample

$$x_t = I(t, x_0, x_1) + \gamma(t)z.$$

Step 2: Learn the fields by regression

The model learns conditional fields from samples of (t, x_t) :

$$\hat{b}(t, x) \approx \mathbb{E}[\dot{x}_t \mid x_t = x], \quad \hat{\eta}_z(t, x) \approx \mathbb{E}[z \mid x_t = x].$$

Equivalently,

$$\hat{s}(t, x) \approx -\gamma(t)^{-1} \hat{\eta}_z(t, x).$$

Step 3: Sampling is chosen after training

Once $\hat{b}$ and $\hat{s}$ are learned, samples from ρ_1 can be generated by integrating an ODE or SDE from the base distribution ρ_0 .

train by regression $\implies$ sample by ODE or SDE.

Algorithmic view: how to train and sample

Step 1: Generate supervised training pairs

Sample

$$x_0 \sim \rho_0, \quad x_1 \sim \rho_1, \quad z \sim \mathcal{N}(0, I_d), \quad t \sim \text{Unif}([0, 1]).$$

Then construct an interpolant sample

$$x_t = I(t, x_0, x_1) + \gamma(t)z.$$

Step 2: Learn the fields by regression

The model learns conditional fields from samples of (t, x_t) :

$$\hat{b}(t, x) \approx \mathbb{E}[\dot{x}_t \mid x_t = x], \quad \hat{\eta}_z(t, x) \approx \mathbb{E}[z \mid x_t = x].$$

Equivalently,

$$\hat{s}(t, x) \approx -\gamma(t)^{-1} \hat{\eta}_z(t, x).$$

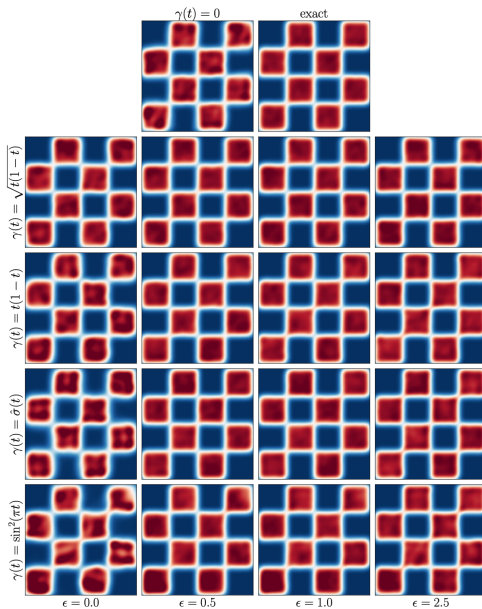
Step 3: Sampling is chosen after training

Once $\hat{b}$ and $\hat{s}$ are learned, samples from ρ_1 can be generated by integrating an ODE or SDE from the base distribution ρ_0 .

$$\text{train by regression} \implies \text{sample by ODE or SDE.}$$

Numerical results

Numerical results: 2D density estimation



Numerical results: 2D density estimation

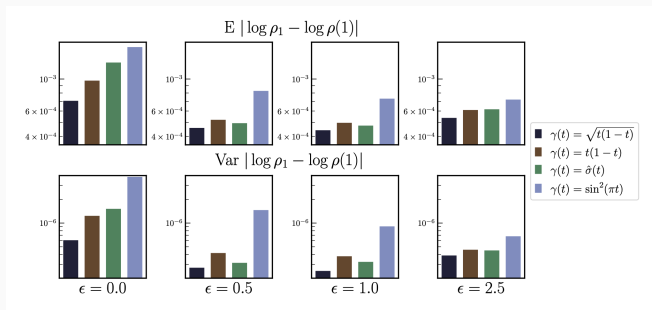


Figure 3: Effect of the interpolant noise $\gamma(t)$ and sampling noise ϵ on sample quality.

Main observation

For this 2D target, SDE sampling is empirically better than ODE sampling for most choices of $\gamma(t)$, but the gap depends strongly on the interpolant design.

Image generation: one-sided interpolants

- The authors test one-sided interpolants on the 128×128 Oxford Flowers dataset.

$$x_t = \cos\left(\frac{\pi t}{2}\right) z + \sin\left(\frac{\pi t}{2}\right) x_1.$$

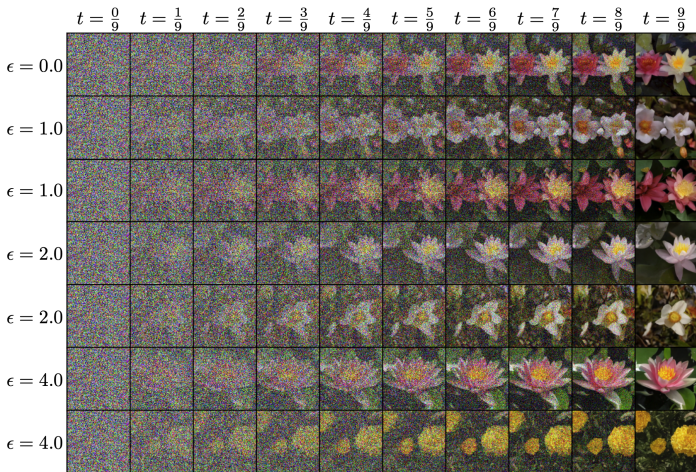


Image generation: diversity and resampling

Figure 14: no simple memorization

The generated flower is compared with its five nearest neighbors in the training set under ℓ_1 distance.

The nearest training images look visually different, suggesting that the model is not simply copying training samples.

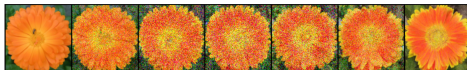


Figure 15: mirror interpolant

For the mirror interpolant,

$$x_t = x_1 + \gamma(t)z,$$

the ODE maps the input image back to itself, while the SDE can resample it into a nearby but different flower.



Final takeaways and limitations

Three main contributions

1. **Unification.** Stochastic interpolants provide a common framework for flow models, diffusion models, denoising methods, stochastic localization, rectified flows, and bridge-based generative modeling.
2. **Flexibility.** The probability path $\rho(t)$ can be designed independently from the sampling dynamics. After choosing an interpolant, one can sample with either an ODE or an SDE.
3. **Learnability.** The required fields, such as the velocity $b(t, x)$, score $s(t, x)$, and denoiser $\eta_z(t, x)$, can be learned through quadratic objectives, making the framework compatible with empirical risk minimization.

Main caveat

The framework introduces many design degrees of freedom:

interpolant, coupling, $\gamma(t)$, $\epsilon(t)$.

It clarifies the design space, but it does not eliminate the need to choose a good design.