Computer Methods in Applied Mechanics and Engineering 461 (2026) 119171

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Generative AI for material design: A mechanics perspective from burgers to matter

Vahidullah Tac, Ellen Kuhl *

Department of Mechanical Engineering, Stanford University, Stanford, CA, United States

ARTICLE INFO

Dataset link: https://github.com/LivingMatterLab/AI4Food

Keywords:
Generative artificial intelligence
Diffusion models
Discrete diffusion
Continuous diffusion
Machine learning

ABSTRACT

Generative artificial intelligence offers a new paradigm to design matter in high-dimensional spaces. However, its underlying mechanisms remain difficult to interpret and limit adoption in computational mechanics. This gap is striking because its core tools – diffusion, stochastic differential equations, and inverse problems – are fundamental to the mechanics of materials. Here we show that diffusion-based generative AI and computational mechanics are rooted in the same principles. We illustrate this connection using a three-ingredient burger as a minimal benchmark for material design in a low-dimensional space, where both forward and reverse diffusion admit analytical solutions: Markov chains with Bayesian inversion in the discrete case and the Ornstein–Uhlenbeck process with score-based reversal in the continuous case. In both cases, forward diffusion adds noise to degrade structure, while reverse diffusion recovers structure from noise. We extend this framework to a high-dimensional design space with 146 ingredients and 8.9×10^{43} possible configurations, where analytical solutions become intractable. We therefore learn the discrete and continuous reverse processes using neural network models that infer inverse dynamics from data. We train the models on only 2260 recipes and generate one million samples that capture the statistical structure of the data, including ingredient prevalence and quantitative composition. We further generate five new burgers and validate them in a blinded restaurant-based sensory study with n = 101 participants, where three of the AI-designed burgers outperform the classical Big Mac in overall liking, flavor, and texture. These results establish diffusion-based generative modeling as a physically grounded approach to design in high-dimensional spaces. They position generative AI as a natural extension of computational mechanics, with applications from burgers to matter, and establish a path towards data-driven, physics-informed generative design. Our source code, data, and examples are available at https://github.com/LivingMatterLab/AI4Food.

1. Motivation

The modern hamburger emerged in the late 19th century as a simple combination of ground meat and bread [1]. Since then, its formulation has evolved through incremental variation; yet, its core design remains largely unchanged [2]. Despite its global popularity – with more than 50 billion hamburgers consumed annually in the United States alone [3] – the massive combinatorial space of possible ingredient configurations remains largely unexplored. This challenge extends far beyond food: the design of matter

* Corresponding author.
E-mail address: ekuhl@stanford.edu (E. Kuhl).

https://doi.org/10.1016/j.cma.2026.119171
Received 3 April 2026; Received in revised form 11 June 2026; Accepted 12 June 2026

and materials draws from vast libraries of candidate components that generate high-dimensional combinatorial spaces beyond systematic exploration [4]. This gap between vast design potential and limited human search motivates a new approach to discovery.

Generative AI for discovery. Generative artificial intelligence is redefining how we explore high-dimensional design spaces and discover new materials across science and engineering [5,6]. Scientists now apply these methods across domains such as computational drug design [7], molecular generation [8], and materials discovery [9], where they propose new candidates that extend far beyond existing datasets [10]. These problems share a fundamental challenge: the underlying design spaces grow combinatorially, while available training data remain sparse [11]. Diffusion-based generative models address this challenge by introducing a forward process that progressively randomizes data and a reverse process that reconstructs structure from noise [12]. This forward–reverse formulation enables controlled sampling of complex distributions and produces new candidates that remain consistent with observed data, while exploring previously unseen regions of the design space [13]. Diffusion models were originally derived from non-equilibrium statistical physics [11]. Our objective is *not* to re-establish this origin, but to translate these concepts into the language of computational mechanics. This perspective casts generative modeling in terms of stochastic processes, transport, and inverse problems, and makes these methods directly accessible to the mechanics community.

A mechanics perspective. From a mechanics perspective, diffusion models take a familiar form: stochastic dynamical systems governed by drift–diffusion equations in high-dimensional state spaces [14]. The forward process adopts *Fokker–Planck diffusion* of probability densities [15,16], $\partial p/\partial t = -\nabla\cdot\boldsymbol{j}$, where the flux, $\boldsymbol{j} = p\,\boldsymbol{b} - \frac{1}{2}\,\boldsymbol{D}\cdot\nabla p$, governs how the probability density p drifts and diffuses under the velocity $\boldsymbol{b}$ that controls deterministic transport, and the diffusion tensor $\boldsymbol{D}$ that controls stochastic spreading. Without the drift, this reduces to classical diffusion along concentration gradients [17], similar to *Fickian diffusion* of concentrations [18] or *Fourier diffusion* of heat [19], $\partial p/\partial t = -\nabla\cdot\boldsymbol{j}$, where the mass or heat flux, $\boldsymbol{j} = -\boldsymbol{D}\cdot\nabla p$, governs how concentration or heat p diffuse under the diffusivity or conductivity tensor $\boldsymbol{D}$. In the discrete setting, the same evolution reduces to a random walk on a graph, $\mathrm{d}p_i/\mathrm{d}t = \sum_j \llcorner_{ij}\, p_j$, where $\llcorner$ denotes the *graph Laplacian* or *Kirchhoff matrix* [20], which governs the diffusion of probability mass across neighboring nodes p_i and p_j of the graph [21]. The reverse process transports probability mass up density gradients through a learned drift term and parallels driven transport in phase-separating systems such as *Cahn–Hilliard diffusion* [22]. The flux of these systems, $\boldsymbol{j} = -\boldsymbol{D}\cdot\nabla\mu$, derives from *Onsager's variational principle* [23], which leads to linear flux-force relations driven by gradients of the chemical potential, $\mu = \delta\mathcal{F}/\delta p$, rather than by concentration gradients ∇p alone, to enable unmixing and uphill transport. This perspective places generative diffusion models within the well-established framework of gradient flows, variational principles, and entropy-driven evolution, and highlights that the mathematical tools of computational mechanics already provide a natural language to analyze, reinterpret, and extend diffusion models in generative AI [24–27].

Food as a model system. Food represents a high-impact societal challenge for generative AI, as it forms a class of complex, multicomponent materials in which composition, structure, and processing jointly determine mechanical response and sensory perception [28–31]. Food scientists are beginning to use machine learning to capture relationships between ingredients, texture, and consumer preference, which enables data-driven optimization of food products [32–37]. Generative models extend this approach by proposing entirely new formulations, which opens the door to systematically explore of the vast combinatorial ingredient space [38]. A new generation of AI-driven food technology companies, including pioneers such as NotCo, is already translating these ideas into practice by designing plant-based foods that replicate the sensory properties of animal products [39,40]. These developments position food as an ideal testbed for generative design in high-dimensional material systems [26].

Burgers as a benchmark. To illustrate these ideas, we introduce burgers as a minimal yet expressive model system for generative design. We begin with a three-ingredient problem, in which each burger consists of bun, patty, and cheese, and train diffusion models on only two examples, hamburger and cheeseburger, to illustrate how forward and reverse diffusion generate new combinations in a finite state space. From a mechanics perspective, this system defines a *low-dimensional* phase space in which we can explicitly analyze and illustrate discrete and continuous diffusion processes, and use it as a transparent setting to study transport, entropy, and structure formation. We then extend this framework to a realistic setting with 146 ingredients and 2260 training recipes, where the design space becomes astronomically large and cannot be explored exhaustively. Within this *high-dimensional* setting, diffusion models enable principled sampling of novel burgers that remain consistent with the training data, while exploring new regions of ingredient space. This progression – from a simple discrete system to a complex continuous problem – establishes burgers as a canonical model for generative design and provides a concrete entry point for applying the tools of computational mechanics to high-dimensional discovery problems.

Outline. Our objective is to establish diffusion models for generative material design, first through a minimal benchmark problem, a three-ingredient burger for which analytical solutions are possible, and then through a real-world problem with more than a hundred possible ingredients for which generative modeling becomes imperative. Section 2 introduces *discrete diffusion* for ingredient selection using the three-ingredient burger model system, characterized by the discrete ingredient vector $\boldsymbol{x} = [\, x_{\text{bun}}, x_{\text{patty}}, x_{\text{cheese}} \,] \in \{0,1\}^3$. Section 3 extends this formulation to *continuous diffusion* for ingredient quantification using the same three-ingredient model system, now characterized by the continuous weight vector $\boldsymbol{w} = [\, w_{\text{bun}}, w_{\text{patty}}, w_{\text{cheese}} \,] \in \mathbb{R}^3$. Section 4 generalizes both formulations from the illustrative three-ingredient space to a real-world 146-ingredient space and discusses the challenges associated with high-dimensional generative material design. All three sections share a common structure: they begin with *forward diffusion*, which transforms structure into noise, proceed with *reverse diffusion*, which reconstructs structure from noise, highlight *illustrative examples* of forward and reverse diffusion, and conclude with the *generation of new designs* by sampling from the diffusion model. Section 5 concludes with a discussion of the results and outlines key challenges and opportunities for computational mechanics in generative AI for materials.

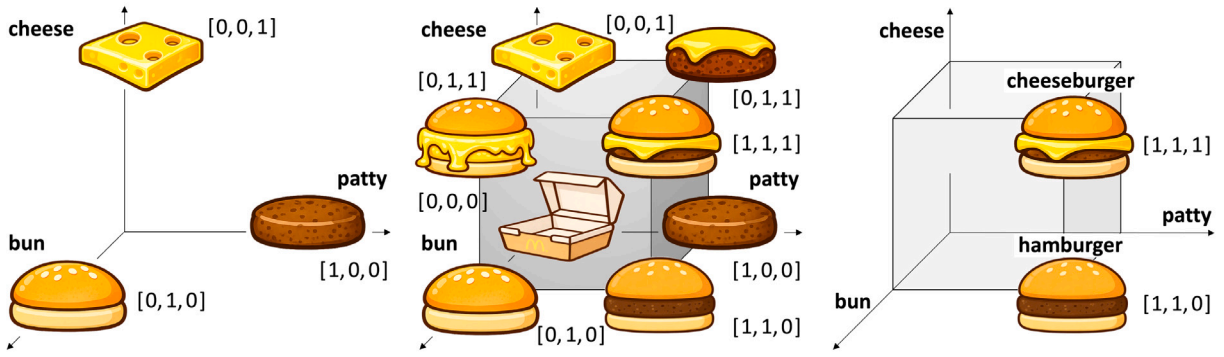

Fig. 1. Three-ingredient burger problem. Three-ingredient space $[\, x_{\text{bun}}, x_{\text{patty}}, x_{\text{cheese}} \,]$ with patty, bun, and cheese (left); with each ingredient either present or absent, $\boldsymbol{x} = [\, x_{\text{bun}}, x_{\text{patty}}, x_{\text{cheese}} \,] \in \{0,1\}^3$, generating $2^3 = 8$ eight possible burgers (middle); training data with cheeseburger $\boldsymbol{x}_1 = [\,1,1,1\,]$ and hamburger $\boldsymbol{x}_2 = [\,1,1,0\,]$ (right).

2. Discrete diffusion

We illustrate the concept of *discrete diffusion* through a minimal benchmark problem: *ingredient selection* for a three-ingredient burger. For this example, we introduce three binary ingredients, bun, patty, and cheese, that are either present or not,

$$\boldsymbol{x} = [\, x_{\text{bun}}, x_{\text{patty}}, x_{\text{cheese}} \,] \;\in\; \{0,1\}^3 \,, \tag{1}$$

Combinatorics introduces a total of $2^3 = 8$ possible burgers, a bun with patty with cheese, a bun with a patty, a bun with cheese, a patty with cheese, just a bun, patty, or cheese, or no ingredients at all (Fig. 1).

Training data. We assume the training data contain two burgers,

$$\boldsymbol{x}_1 = [\,1,1,1\,] \quad \text{and} \quad \boldsymbol{x}_2 = [\,1,1,0\,], \tag{2}$$

a cheeseburger, i.e., a bun with patty with cheese, and a hamburger, i.e., a bun with patty. We assume that both burgers are equally present with an empirical data distribution,

$$p_{\text{data}}(\boldsymbol{x}) = \tfrac{1}{2}\,\delta(\boldsymbol{x} - \boldsymbol{x}_1) + \tfrac{1}{2}\,\delta(\boldsymbol{x} - \boldsymbol{x}_2)\,, \tag{3}$$

where δ denotes the Kronecker delta, such that $p_{\text{data}}(\boldsymbol{x}_1) = \frac{1}{2}$ and $p_{\text{data}}(\boldsymbol{x}_2) = \frac{1}{2}$ and $p_{\text{data}}(\boldsymbol{x}) = 0$ otherwise.

Structure. From the empirical data distribution $p_{\text{data}}(\boldsymbol{x})$, we compute the marginal probabilities P of each ingredient,

$$\mathsf{P}(x_{\text{bun}} = 1) = 1 \quad \text{and} \quad \mathsf{P}(x_{\text{patty}} = 1) = 1 \quad \text{and} \quad \mathsf{P}(x_{\text{cheese}} = 1) = \tfrac{1}{2}. \tag{4}$$

This means that, under the data distribution, the marginal probability that the bun is present is one, that the patty is present is one, and that the cheese is present is one half: Bun and patty are *deterministic*, while cheese is *stochastic*.

Entropy. The *Shannon entropy* quantifies the uncertainty associated with the probability distribution p [41],

$$H(p) = -\textstyle\sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log p(\boldsymbol{x}). \tag{5}$$

For the empirical data distribution, the probability mass is equally split between two states, the bun with patty with cheese, $p_{\text{data}}(\boldsymbol{x}_1) = \frac{1}{2}$ and the bun with patty, $p_{\text{data}}(\boldsymbol{x}_2) = \frac{1}{2}$, thus,

$$H(p_{\text{data}}) = -2\,[\;\tfrac{1}{2}\;\log(\tfrac{1}{2})\;] = \log(2). \tag{6}$$

This entropy reflects the fact that the dataset contains uncertainty along only one coordinate, cheese, while two other coordinates, bun and patty, are fixed. If all eight states were equally likely, $p_{\text{data}}(\boldsymbol{x}) = \frac{1}{8}$ for all $\boldsymbol{x}$, maximum entropy would occur, with

$$H_{\text{max}} = -8\,[\;\tfrac{1}{8}\;\log(\tfrac{1}{8})\;] = \log(8). \tag{7}$$

This implies that the data distribution has a significantly lower entropy than the uniform distribution, $H(p_{\text{data}}) < H_{\text{max}}$. Forward diffusion will gradually increase entropy, and drive the distribution from $\log(2)$ towards $\log(8)$.

Geometric interpretation. *The three binary ingredients define a three-dimensional discrete state space* $\{0,1\}^3$*, and the eight possible burgers correspond to the vertices of the unit cube* $[\,0,1\,]^3$*. In this discrete space, probability mass is initially concentrated on two adjacent vertices of a single edge, rather than spread across all eight vertices of the cube. Although the state space is three-dimensional, the dataset varies only along one coordinate, cheese, while the other two coordinates, bun and patty, are fixed. Forward diffusion will gradually move the probability mass from these two vertices towards all other vertices of the cube* (Fig. 1).

2.1. Forward diffusion

The forward diffusion process gradually destroys structure by randomly perturbing each ingredient. Specifically, at each time step, every ingredient, { bun, patty, cheese }, flips independently with a probability β. A flip means that a present ingredient becomes absent, or an absent ingredient becomes present (Fig. 2).

Single-ingredient transition. For a single binary ingredient $x_i \in \{0, 1\}$, the transition matrix $\mathbf{Q}_1$ governs the transition from one time step to the next,

$$\mathbf{Q}_1 = \begin{bmatrix} [\,1-\beta\,] & \beta \\ \beta & [\,1-\beta\,] \end{bmatrix}. \tag{8}$$

Here, $\beta \in (0, 1)$ is the per-ingredient flip probability, which controls the strength of the diffusion level. With a probability $[\,1-\beta\,]$, the ingredient remains unchanged; with a probability β, the ingredient flips. For small β, ingredients flip slowly, structure decays gradually, and mixing is slow; for large β, ingredients flip frequently, structure disappears quickly, and the ingredient list converges rapidly to a uniform distribution.

All-ingredient transition. Because all three ingredients flip *independently*, the full transition matrix for the burger is the tensor product,

$$\mathbf{Q}_3 = \mathbf{Q}_1 \otimes \mathbf{Q}_1 \otimes \mathbf{Q}_1 \,. \tag{9}$$

The matrix $\mathbf{Q}_3$ defines a *Markov chain* on the eight vertices of the cube $\{0, 1\}^3$. Here we model diffusion through a discrete-time Markov chain, because it defines a simple stochastic process on a finite state space with analytically tractable forward transitions and exact reverse dynamics via Bayes' theorem. We define the forward diffusion process through a transition kernel which independently flips each ingredient with probability β,

$$p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \prod_{i=1}^{3} [\,\beta\,[\,1-\delta_{\boldsymbol{x}_{t,i},\boldsymbol{x}_{t-1,i}}\,][\,1-\beta\,]\,\delta_{\boldsymbol{x}_{t,i},\boldsymbol{x}_{t-1,i}}\,], \tag{10}$$

where δ denotes the Kronecker delta, which equals one if the ingredient state remains unchanged and zero if it flips.

One-step transition probability. We can introduce the explicit one-step transition probability of the forward diffusion Markov chain (10),

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \beta^{d(\boldsymbol{x},\boldsymbol{y})}\,[\,1-\beta\,]^{3-d(\boldsymbol{x},\boldsymbol{y})}, \tag{11}$$

where $d(\boldsymbol{x}, \boldsymbol{y})$ is the *Hamming distance*, the number of ingredients by which two burgers $\boldsymbol{x}$ and $\boldsymbol{y}$ differ. For a three-ingredient burger, within one diffusion step, the probabilities of no flip, one flip, two flips, and all three flips are

$$\mathsf{P}(d=0|\,\boldsymbol{x}) = [\,1-\beta\,]^3 \quad \mathsf{P}(d=1|\,\boldsymbol{x}) = 3\beta\,[\,1-\beta\,]^2 \quad \mathsf{P}(d=2|\,\boldsymbol{x}) = 3\beta^2[\,1-\beta\,] \quad \mathsf{P}(d=3|\,\boldsymbol{x}) = \beta^3\,. \tag{12}$$

For the example of a moderate flip probability of β=0.025, the flip probabilities would be $\mathsf{P}(d=0|\,\boldsymbol{x}) = 0.92686$ and $\mathsf{P}(d=1|\,\boldsymbol{x}) = 0.07130$ and $\mathsf{P}(d=2|\,\boldsymbol{x}) = 0.00183$ and $\mathsf{P}(d=3|\,\boldsymbol{x}) = 0.00002$. This implies that, in one step, at a probability of 0.92686, the burger is usually preserved, but occasionally loses or gains ingredients. Over repeated steps, these random flips accumulate.

Evolution of probability distribution. To calculate the evolution of the distribution, we introduce $p_t(\boldsymbol{x})$, the probability distribution at time t, which evolves according to

$$p_{t+1}(\boldsymbol{y}) = \textstyle\sum_{\boldsymbol{x}} p_t(\boldsymbol{x})\, p(\boldsymbol{y} \mid \boldsymbol{x}). \tag{13}$$

This equation expresses the probability of burger $\boldsymbol{y}$ at time $(t+1)$ as a sum over all possible previous burgers $\boldsymbol{x}$, weighted by their probability $p_t(\boldsymbol{x})$ and the transition probability $p(\boldsymbol{y} \mid \boldsymbol{x})$. It defines a discrete conservation law for probability mass on the state space, analogous to transport equations in continuum mechanics, and describes the evolution of the distribution at the ensemble level.

Analytical solution. The forward diffusion process admits a closed-form solution. For a general initial distribution $p_0(\boldsymbol{x})$, the solution follows by linear superposition, $p_t(\boldsymbol{x}) = \sum_{\boldsymbol{x}_0} p_0(\boldsymbol{x}_0)\, p_t(\boldsymbol{x} \mid \boldsymbol{x}_0)$. For an initial configuration $\boldsymbol{x}_0$, like in our three-ingredient burger problem, the distribution at time t depends only on the Hamming distance $d(\boldsymbol{x}_0, \boldsymbol{x})$ and takes the following explicit form,

$$p_t(\boldsymbol{x} \mid \boldsymbol{x}_0) = q_t^{\,d(\boldsymbol{x}_0,\boldsymbol{x})}\,[\,1-q_t\,]^{3-d(\boldsymbol{x}_0,\boldsymbol{x})} \qquad \text{with} \qquad q_t = \tfrac{1}{2}\,[\,1-[\,1-2\beta\,]^t\,], \tag{14}$$

where q_t denotes the cumulative flip probability after t steps. This implies that in practice, we can simulate the evolution of probabilities in Eq. (13) simply by sampling trajectories of the underlying Markov chain, which generates stochastic paths from $\boldsymbol{x}_0 \to \boldsymbol{x}_1 \to \ldots \to \boldsymbol{x}_T$ through independent Bernoulli updates at each time step.

Geometric interpretation. *Forward diffusion is a symmetric random walk on the cube graph. The eight burgers represent the vertices of the cube. The edges connect burgers that differ in exactly one ingredient. Probability flows along the edges of the cube. Because transitions are symmetric, no burger is preferred. This process gradually spreads probability mass from the two training vertices to all eight vertices of*

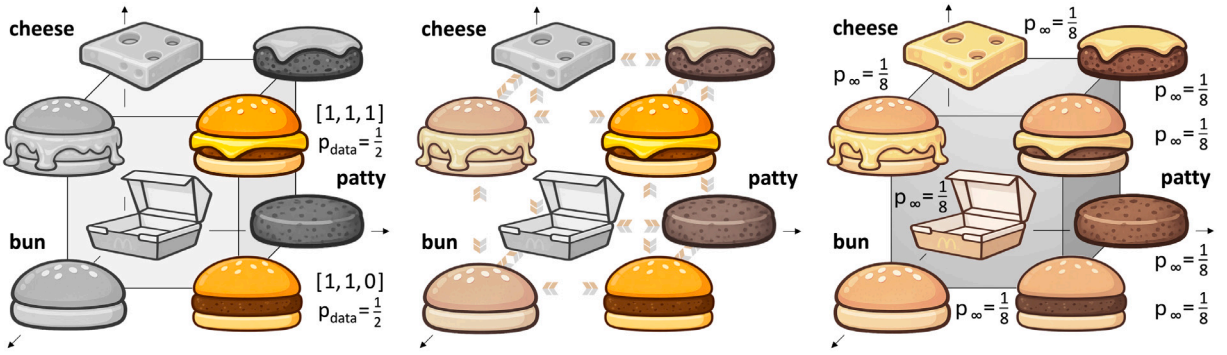

Fig. 2. Discrete modeling of forward diffusion. Three-ingredient space with eight possible burgers with training data, cheeseburger $\boldsymbol{x}_1 = [\,1, 1, 1\,]$ and hamburger $\boldsymbol{x}_2 = [\,1, 1, 0\,]$ with equal probabilities $p_{\text{data}}(\boldsymbol{x}_1) = 0.50$ and $p_{\text{data}}(\boldsymbol{x}_2) = 0.50$, highlighted in color (left); forward diffusion with independent equal-probability flipping of the three ingredients gradually diffuses probabilities across the cube (middle); converged state of maximum entropy with equal probabilities $p_\infty = \frac{1}{8}$ across all eight burgers (right).

the cube. Over time, this mixing destroys the original structure that the bun and patty are always present and the distribution approaches uniformity (Fig. 2).

Entropy and convergence. For $0 < \beta < 1$, every ingredient has a positive probability of flipping β, and a positive probability of remaining unchanged $[\,1 - \beta\,]$. Consequently, from any burger state, it is possible to reach any other burger state in a finite number of steps. The Markov chain defined by $\mathbf{Q}_3$ is therefore *irreducible*. Because self-transitions occur with probability $[\,1 - \beta\,]^3 > 0$, the chain is also *aperiodic*. Hence, the chain is *ergodic* and admits a unique stationary distribution.

Stationary distribution. Since the transition matrix $\mathbf{Q}_3$ is symmetric, all states are treated equally. The stationary distribution must therefore be *uniform*,

$$p_\infty(\boldsymbol{x}) = \tfrac{1}{8} \qquad \boldsymbol{x} \in \{0, 1\}^3. \tag{15}$$

Regardless of the initial distribution, as $t \to \infty$, the stationary distribution will converge to this uniform distribution, $p_t \to p_\infty$. In other words, repeated random ingredient flips eventually make all eight burgers equally likely.

Entropy evolution. Recall that the Shannon entropy is

$$H(p) = -\textstyle\sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log(p(\boldsymbol{x}))\,. \tag{16}$$

Initially, the data distribution is supported on two burgers. At stationarity, all eight burgers are equally likely. For the entropy, this implies,

$$H(p_0) = \log(2) \qquad \text{and} \qquad H(p_\infty) = \log(8) = 3\,\log(2) = H_{\max}. \tag{17}$$

The entropy increases from $\log(2)$ to $\log(8)$, a threefold increase in uncertainty. Forward diffusion increases entropy and converges to the maximum-entropy distribution.

Burger interpretation. *Initially, bun and patty were deterministic, and only cheese varied, $x_{\text{cheese}} = \{0, 1\}$, After sufficient diffusion, bun may be present or absent, $x_{\text{bun}} = \{0, 1\}$, patty may be present or absent, $x_{\text{patty}} = \{0, 1\}$, cheese may be present or absent, $x_{\text{cheese}} = \{0, 1\}$. Forward diffusion forgets structure. All combinations become equally likely. The structured dataset dissolves into complete uncertainty. Forward diffusion spreads the probability mass from the two initial vertices $\boldsymbol{x}_1 = [\,1, 1, 1\,]$ and $\boldsymbol{x}_2 = [\,1, 1, 0\,]$, across the cube, and in the long-time limit, converges to the uniform distribution on all eight vertices* (Fig. 2).

Analogy to mechanics. *We can interpret the discrete forward diffusion process as a random walk on a finite state space, governed by the Markov update, $p_{t+1}(\boldsymbol{y}) = \sum_{\boldsymbol{x}} p_t(\boldsymbol{x})\, P(\boldsymbol{y} \to \boldsymbol{x})$, which is formally analogous to a master equation that describes stochastic hopping between discrete configurations in lattice-based models. In this interpretation, the transition probabilities $P(\boldsymbol{y} \to \boldsymbol{x})$ define local hopping rates between neighboring states. The transition kernel defines a stochastic nearest-neighbor interaction on the cube, where each ingredient behaves like a binary degree of freedom that switches between two states with flip probability β. Repeated application of this operator leads to a progressive spreading of probability mass, and over time, the distribution converges towards a uniform equilibrium that corresponds to a system with no energetic preference among configurations. In this formulation, the generator L acts as a discrete Laplacian on the state graph, and the evolution corresponds to diffusion of probability mass across neighboring configurations* [42].

2.2. Reverse diffusion

While the forward diffusion process *destroys structure* by randomly flipping ingredients, the reverse process aims to *reconstruct structure* from partially corrupted burgers. The reverse process is governed by the time-reversed transition probability $p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$,

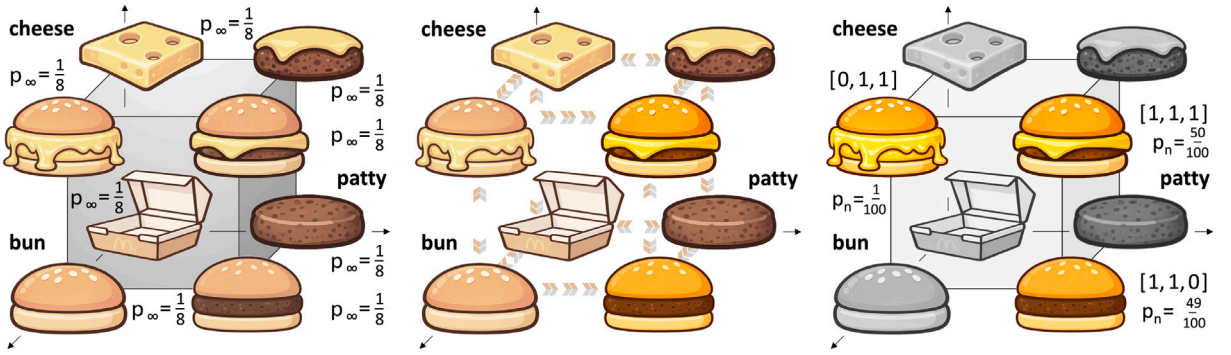

Fig. 3. Discrete modeling of reverse diffusion. Noised data with maximum entropy and equal probabilities $p_\infty = \frac{1}{8}$ of all eight burgers (left); reverse diffusion with probability-weighted flipping of the three ingredients gradually diffuses probabilities against their gradients towards the training set (middle); de-noised data with three possible burgers, cheeseburger $\boldsymbol{x}_1 = [\,1, 1, 1\,]$ and hamburger $\boldsymbol{x}_2 = [\,1, 1, 0\,]$ and newly discovered cheese sandwich $\boldsymbol{x}_3 = [\,0, 1, 1\,]$ with probabilities $p_{\text{data}}(\boldsymbol{x}_1) = 0.50$ and $p_{\text{data}}(\boldsymbol{x}_2) = 0.49$ and $p_{\text{data}}(\boldsymbol{x}_3) = 0.01$, highlighted in color (right).

the probability that the previous burger at time $(t-1)$ was $\boldsymbol{x}_{t-1} \in \{0,1\}^3$ given the observed burger at time t is $\boldsymbol{x}_t \in \{0,1\}^3$. For the three-ingredient burger problem, we can calculate the reverse kernel analytically using *Bayes' theorem*,

$$p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) = \frac{p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})\, p_{t-1}(\boldsymbol{x}_{t-1})}{\sum_{\boldsymbol{x}} p(\boldsymbol{x}_t \mid \boldsymbol{x})\, p_{t-1}(\boldsymbol{x})}, \tag{18}$$

where $p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ is the *posterior*, the probability of the previous state $\boldsymbol{x}_{t-1}$ given the current state $\boldsymbol{x}_t$; $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$ is the *likelihood*, the forward transition probability from Eq. (11); $p_{t-1}(\boldsymbol{x}_{t-1})$ is the *prior*, the distribution of states at time $(t-1)$ obtained from the forward solution in Eqs. (13) and (14); and $\sum_{\boldsymbol{x}} p(\boldsymbol{x}_t \mid \boldsymbol{x})\, p_{t-1}(\boldsymbol{x})$ is the *evidence*, the sum over all possible burger configurations $\boldsymbol{x} \in \{0,1\}^3$ that normalizes the probabilities to sum up to one. For the three-ingredient benchmark, we can evaluate the reverse kernel *exactly*, because the state space contains only $2^3 = 8$ configurations. For the full 146-ingredient problem in Section 4, this exact computation becomes intractable, because the corresponding normalization sum $\sum_{\boldsymbol{x}}$ would run over 2^{146} possible configurations. In this case, we will evaluate the reverse kernel *approximately* by learning a model $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$, for example a neural network with trainable parameters $\boldsymbol{\theta}$. In intuitive terms, if the forward process gradually moves probability mass away from the two training burgers, the reverse process learns how to move probability mass back towards them (Fig. 3).

Numerical reverse sampling. To generate trajectories from the reverse process, we sample sequentially from the conditional distributions $p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ in Eq. (18). Starting from a noisy configuration $\boldsymbol{x}_T$ that we typically draw from an approximately uniform distribution, we iteratively generate a sequence $\boldsymbol{x}_T \to \boldsymbol{x}_{T-1} \to \cdots \to \boldsymbol{x}_0$, by sampling $\boldsymbol{x}_{t-1}$ from $p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ at each step. This procedure defines a stochastic trajectory that progressively reconstructs structure by transporting probability mass from high-entropy states towards the low-entropy data distribution.

Geometric interpretation. *The forward process defines a random walk on the cube: Transitions along the edges are symmetric, all directions are equally likely, and probability spreads uniformly across the cube. The reverse process defines a biased random walk on the cube: Transitions are no longer symmetric, but depend on the data distribution. Edges that point towards high-density vertices,* $[\,1, 1, 0\,]$ *and* $[\,1, 1, 1\,]$*, have a higher transition probability, while edges pointing away from the data manifold have a lower transition probability. Bayes' theorem assigns transition probabilities along the edges of the reverse random walk so that probability mass flows preferentially towards the training vertices,* $[\,1, 1, 0\,]$ *and* $[\,1, 1, 1\,]$*. In the long-time limit, this biased random walk concentrates probability near the initial data support. It reduces entropy and restores structure* (Fig. 3).

2.3. Discrete forward and reverse diffusion

We now illustrate the behavior of discrete diffusion for the eight possible burgers defined by binary ingredient selection (Figs. 4 and 5). The *forward diffusion* problem (11) admits a closed-form solution: For an initial configuration $\boldsymbol{x}_0$, the distribution p_t at time t only depends on the Hamming distance $d(\boldsymbol{x}_0, \boldsymbol{x})$ and takes the explicit form $p_t(\boldsymbol{x} \mid \boldsymbol{x}_0) = q_t^{\,d(\boldsymbol{x}_0,\boldsymbol{x})}[\,1 - q_t\,]^{3-d(\boldsymbol{x}_0,\boldsymbol{x})}$, where $q_t = \frac{1}{2}[\,1 - [\,1 - 2\beta\,]^t\,]$ represents the cumulative flip probability after t diffusion steps. For the *reverse diffusion* problem (18), we generate trajectories Bayes' theorem by sampling sequentially from the posterior transition probabilities $p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$, which we obtain from Bayes' theorem using the forward transition probabilities and marginals, where the dependence on the initial data $\boldsymbol{x}_0$ is implicitly encoded through the forward marginals $p_{t-1}(\boldsymbol{x})$. This procedure produces stochastic paths $\boldsymbol{x}_T \to \boldsymbol{x}_{T-1} \to \cdots \to \boldsymbol{x}_0$ that reconstruct likely ingredient configurations consistent with the forward diffusion process. During forward diffusion (Fig. 4), the model begins with a degenerate distribution concentrated on two equally probable training burgers, cheeseburger and hamburger, with $p_0 = 0.500$ (red lines), and progressively spreads probability mass towards the cheese sandwich, plain bun, cheese patty, and plain patty (orange lines), and with a slight delay towards the plain cheese and the empty burger box (blue lines). Over time, the distribution approaches uniformity with $p_\infty = 0.125$, as evidenced by the convergence of all state probabilities towards the

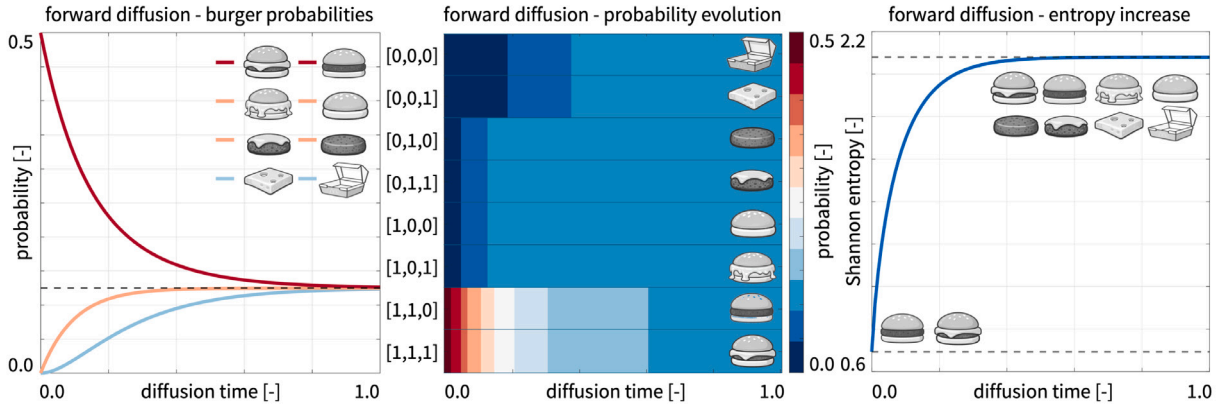

Fig. 4. Forward diffusion in discrete ingredient space. Forward diffusion noises the training data to obtain a uniform distribution. Starting from two training states, cheeseburger and hamburger, probability mass spreads across all states and approaches a uniform distribution as reflected by the convergence of individual state probabilities (left), homogenization of the probability distribution (center), and increase in Shannon entropy (right).

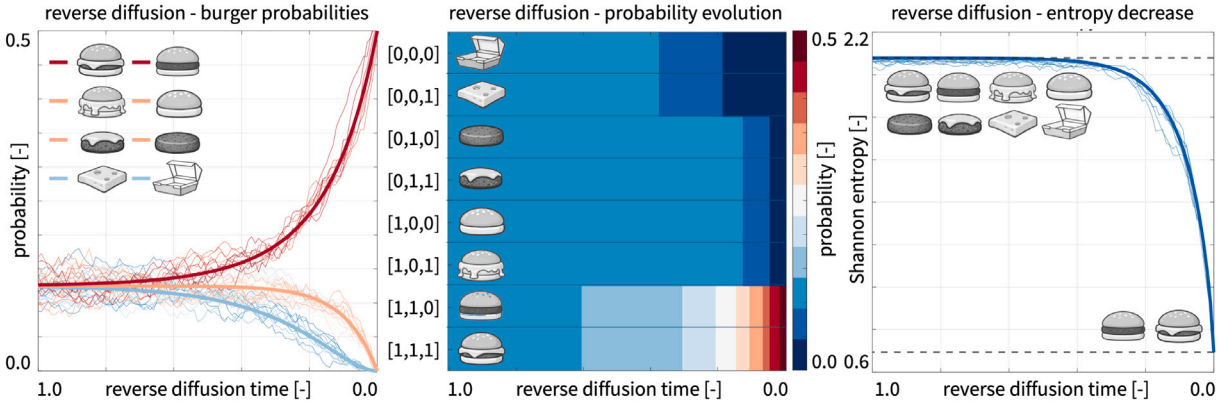

Fig. 5. Reverse diffusion in discrete ingredient space. Reverse diffusion recovers the training distribution from the uniform state. Starting from the noised state, probability mass concentrates back onto the original two burgers as reflected by the divergence of state probabilities (left), re-localization in probability space (center), and rapid entropy decrease (right). Thin lines indicate stochastic realizations; thick lines show ensemble averages.

same value and the corresponding increase in Shannon entropy to $H(p_\infty) = 2.079$. This reflects the role of forward diffusion as a mixing process, which gradually removes information about the initial configuration. During reverse diffusion (Fig. 5), the model inverts the diffusion dynamics. Starting from an approximately uniform distribution with $p_\infty = 0.125$, reverse diffusion progressively concentrates probability mass back onto the training states. This manifests itself in a separation of probabilities towards the cheeseburger and hamburger that converge towards $p_0 = 0.500$ as the entropy decreases towards $H(p_0) = 0.6931$. The thin lines of the individual stochastic trajectories visualize the variability, while the thick lines of the ensemble average follow a smooth and consistent contraction towards the data distribution. Taken together, these results demonstrate that in the discrete setting, diffusion operates over a finite combinatorial state space, where all configurations are accessible during forward diffusion and the learned reverse dynamics selectively reconstruct the observed data modes.

Analogy to mechanics. *The increase of entropy during forward diffusion reflects the loss of information and the approach to equilibrium, consistent with the second law of thermodynamics. Conversely, reverse diffusion reduces entropy by reintroducing structure through a learned drift term and effectively acts as a data-driven inverse process. In this sense, it resembles transport against concentration gradients, ∇p, or uphill diffusion, where probability mass flows towards higher-density regions under the action of an externally learned driving force. This is analogous to driven transport processes in mechanics, such as unmixing processes that restore structure from a mixed state under an external force.*

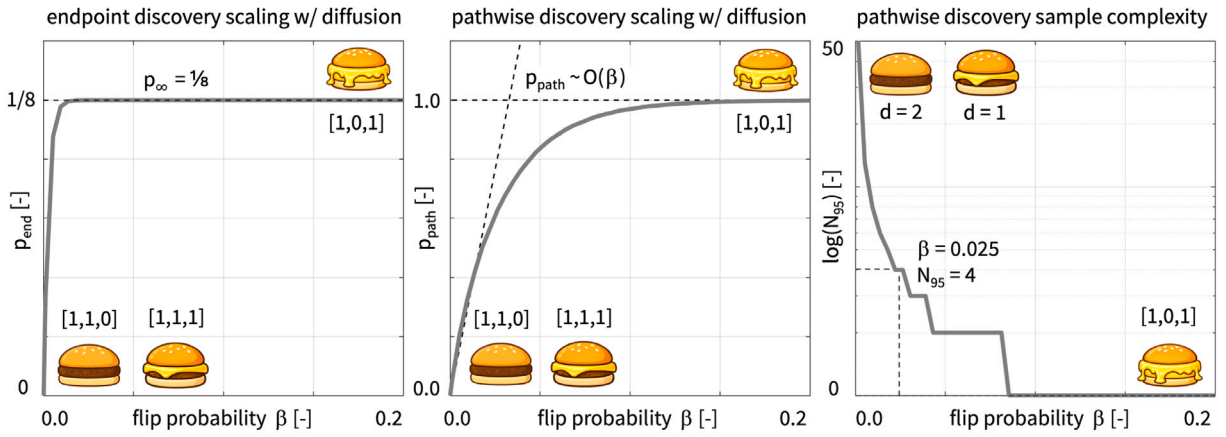

Fig. 6. Generating new burgers by discrete diffusion. Sampling complexity for discovering a new burger, a cheese sandwich $[\,1,0,1\,]$, with Hamming distances of $d = 2$ and $d = 1$ from the training data, hamburger $[\,1,1,0\,]$ and cheeseburger $[\,1,1,1\,]$. Probability of endpoint discovery p_{end} increases with flip probability β and saturates at the uniform limit $p_\infty = \frac{1}{8}$ (left); probability of pathwise discovery p_{path} increases from an initial small-β scaling, $p_{\text{path}} \approx \mathcal{O}(\beta)$, and approach unity as trajectories explore the full state space (middle); number of sample trajectories required for 95% discovery N_{95} decreases rapidly with increasing flip probability β (right).

2.4. Generating new burgers by discrete diffusion

We next quantify the ability of discrete diffusion to generate burgers not present in the training set. As a representative example, we consider the cheese sandwich $[1,0,1]$, which differs from the training burgers by one ingredient flip from the cheeseburger, $d = 1$, and two flips from the hamburger, $d = 2$. We report sampling complexity via the probabilities p_{path} and p_{end} that quantify the likelihood that a single trajectory passes through or ends at the defined target burger, and via the number of independent samples required to achieve 95% probability of discovery $N_{95} = [\log(0.05)/\log(1-p)]$ after $T = 100$ diffusion steps (Fig. 6). Discovery probability increases rapidly with increasing flip probability β : For small β, discovery is limited by the probability of rare multi-bit flips, while for large β, the process approaches uniform sampling over the state space. As a results, the endpoint probability p_{end} increases with flip probability β and approaches the uniform value $p_\infty = \frac{1}{8}$ that reflects complete mixing over the eight possible burgers. This discrete solution agrees with the analytical solution, $p_{\text{end}} = \frac{1}{2}\,q\,[1-q]$ with $q = \frac{1}{2}\,[1-[1-2\,\beta]^T\,]$, where T is the number of diffusion steps. In contrast, the pathwise probability p_{path} initially obeys small-β scaling, $p_{\text{path}} \approx \mathcal{O}(\beta)$, but then rises to approaches unity, since the cheese sandwich can be encountered at any intermediate step along a trajectory. The corresponding discovery cost N_{95} decreases sharply with increasing flip probability β.

3. Continuous diffusion

We now move from discrete diffusion for ingredient selection to *continuous diffusion* for *ingredient quantification*. Instead of binary ingredients, we consider real-valued ingredient weights,

$$\boldsymbol{w}^* = [\,w^*_{\text{bun}}, w^*_{\text{patty}}, w^*_{\text{cheese}}\,] \in \mathbb{R}^3, \tag{19}$$

where each component denotes the weight in grams of the ingredient. For convenience, we normalize all weights by the weights of a normal sized bun $w^0_{\text{bun}} = 55\,\text{g}$, a small patty $w^0_{\text{patty}} = 45\,\text{g}$, and a slice of cheese $w^0_{\text{cheese}} = 14\,\text{g}$, and translate the burger into a coordinate system with the cheeseburger at the origin,

$$\boldsymbol{w} = [\,w_{\text{bun}}, w_{\text{patty}}, w_{\text{cheese}}\,] = [\,w^*_{\text{bun}}/w^0_{\text{bun}}, w^*_{\text{patty}}/w^0_{\text{patty}}, w^*_{\text{cheese}}/w^0_{\text{cheese}}\,] - [\,1,1,1\,] \in \mathbb{R}^3\,. \tag{20}$$

In these two coordinate systems, the cheeseburger is represented as $\boldsymbol{w}^*_1 = [\,55\,\text{g}, 45\,\text{g}, 14\,\text{g}\,]$ and $\boldsymbol{w}_1 = [\,0,0,0\,]$ and the hamburger is $\boldsymbol{w}^*_2 = [\,55\,\text{g}, 45\,\text{g}, 0\,\text{g}\,]$ and $\boldsymbol{w}_2 = [\,0,0,-1\,]$.

Training data. The training data consist of N burgers, $\{\boldsymbol{w}_i\}_{i=1}^N$, each representing a set of ingredient weights. For this case, the empirical data distribution is $p_{\text{data}}(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^N \delta(\boldsymbol{w} - \boldsymbol{w}_i)$, where, in the continuous case, δ is the Dirac delta distribution. Here, similar to the discrete case, we assume that the training data consist of two burgers,

$$\boldsymbol{w}_1 = [\,0,0,0\,] \quad \text{and} \quad \boldsymbol{w}_2 = [\,0,0,-1\,], \tag{21}$$

a cheeseburger with a standard size bun, patty, and cheese, and a hamburger with a standard size bun and patty, but no cheese, with equal probabilities,

$$p_{\text{data}}(\boldsymbol{w}) = \tfrac{1}{2}\delta(\boldsymbol{w} - \boldsymbol{w}_1) + \tfrac{1}{2}\delta(\boldsymbol{w} - \boldsymbol{w}_2)\,, \tag{22}$$

such that $p_{\text{data}}(\boldsymbol{w}_1) = \frac{1}{2}$ and $p_{\text{data}}(\boldsymbol{w}_2) = \frac{1}{2}$.

Geometric interpretation. *The three continuous ingredient weights define a three-dimensional state space $\mathbb{R}^3$, with an infinite number of possible burgers, where the cheeseburger defines the origin, burgers with lower ingredient weights like the cheese-less hamburger have negative coordinates, and burgers with higher ingredient weights like the two-patty Mc Double have positive coordinates. Unlike in the discrete case, the state space is now continuous. The discrete corners of the cube are replaced by the entire three-dimensional Euclidean space. Each burger is now a point in $\mathbb{R}^3$. The training data form a low-dimensional cloud within this space, in our example, represented by only two points.*

3.1. Forward diffusion

Forward diffusion gradually destroys structure by noising the training data. It modifies each burger point by random Gaussian noise, and at the same time, gently pulls it towards the origin. Specifically, we introduce continuous forward diffusion through a stochastic differential equation,

$$\mathrm{d}\boldsymbol{w}_t = \boldsymbol{f}(\boldsymbol{w}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{B}_t, \tag{23}$$

where $\boldsymbol{w}_t \in \mathbb{R}^3$ is the ingredient-weight vector at diffusion time t, with components $\boldsymbol{w}_t = [\, w_{\text{bun}}(t),\, w_{\text{patty}}(t),\, w_{\text{cheese}}(t)\,]$, $\mathrm{d}\boldsymbol{w}_t$ is the *stochastic differential* that represents the infinitesimal random increment of the process that consists of a deterministic drift and a stochastic fluctuation, $\boldsymbol{f}(\boldsymbol{w}_t, t)$ is the drift vector that defines the systematic trend, $g(t)$ is the diffusion coefficient that defines the noise magnitude, and $\mathrm{d}\boldsymbol{B}_t$ is the vector-valued stochastic differential of Brownian motion. The *Brownian motion* of the continuous diffusion case represents a continuous-time limit of the *multinomial transitions* or ingredient flipping of the discrete diffusion case.

Ornstein–Uhlenbeck forward process. Here we model diffusion through a variance-preserving *Ornstein–Uhlenbeck process* [43], because it provides a linear Gaussian diffusion with a closed-form solution that enables analytically tractable forward and reverse processes,

$$\mathrm{d}\boldsymbol{w}_t = -\tfrac{1}{2}\,\beta(t)\,\boldsymbol{w}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{B}_t\,. \tag{24}$$

The first term, $-\frac{1}{2}\beta(t)\boldsymbol{w}_t$, is the *drift* that removes information by contracting all weights towards zero. The second term, $\sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{B}_t$, is the *stochastic diffusion* that increases entropy by injecting isotropic Gaussian noise to spread the weights outward. The parameter $\beta(t) > 0$ is the noise schedule or diffusion rate that controls both the strength of fluctuations and the timescale of drift. It increases as time progresses, from an initial weak contraction and weak noise towards a strong contraction and strong noise. This stochastic differential equation has the following closed-form conditional distribution,

$$p(\boldsymbol{w}_t \,|\, \boldsymbol{w}_0) = \mathcal{N}(\,\boldsymbol{w}_t \,|\, \mu(t)\,\boldsymbol{w}_0, \sigma(t)\boldsymbol{I}\,)\,, \tag{25}$$

with

$$\mu(t) = \exp(-\tfrac{1}{2}\alpha(t)) \quad \text{and} \quad \sigma(t) = 1 - \exp(-\alpha(t)) \quad \text{and} \quad \alpha(t) = \textstyle\int_0^t \beta(s)\,\mathrm{d}s\,, \tag{26}$$

where $\mu(t)$ is the signal attenuation factor that determines how much of $\boldsymbol{w}_0$ remains, $\sigma(t)$ is the noise variance that is accumulated by time t, and $\alpha(t)$ is the integrated noise.

Analogy to mechanics. *The Ornstein–Uhlenbeck process is equivalent to a linear Langevin equation that governs stochastic relaxation under competing dissipation and fluctuations. At the ensemble level, the process satisfies a Fokker–Planck equation that describes probability transport under linear drift and isotropic diffusion* [15,16]. *Here, β acts as a relaxation rate that controls both noise intensity and drift strength, and sets the timescale of mixing. Increasing β results in faster exploration of the state space over finite time horizons, while leaving the equilibrium distribution unchanged.*

One-step all-weight diffusion. As illustrative example, we assume a three-ingredient burger, with initial ingredients

$$\boldsymbol{w}_0^* = [\,55\,\text{g}, 45\,\text{g}, 0\,\text{g}\,] \quad \text{or} \quad \boldsymbol{w}_0 = [\,0, 0, -1\,]\,, \tag{27}$$

which corresponds to a hamburger with a 55 g bun and a 45 g patty. For Ornstein–Uhlenbeck forward diffusion, the noisy burger is $\boldsymbol{w}_t = \mu(t)\,\boldsymbol{w}_0 + \sqrt{\sigma(t)}\,\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$. We choose a representative diffusion time such that $\alpha(t) = \log(2)$, and assume a Gaussian noise realization of $\boldsymbol{\epsilon} = [\,0, \sqrt{2}, 1\,]$. For this example, the signal attenuation factor $\mu(t)$ and the noise variance $\sigma(t)$ become, $\mu(t) = 0.707$ and $\sigma(t) = 0.5$. The resulting mean, $\mu(t)\,\boldsymbol{w}_0 = [\,0, 0, -0.707\,]$, and injected noise, $\sqrt{\sigma(t)}\,\boldsymbol{\epsilon} = [\,0.000, 1.000, 0.707\,]$, produce a noisy burger with the following weights,

$$\boldsymbol{w}_t = [\,0, 1, 0\,] \quad \text{or} \quad \boldsymbol{w}_t^* = [\,55\text{g}, 90\text{g}, 14\text{g}\,]. \tag{28}$$

The initial hamburger $\boldsymbol{w}_0 = [\,0, 0, -1\,]$ turns into a Mc Double $\boldsymbol{w}_t = [\,0, 1, 0\,]$, with a 55 g bun and two 45 g patties and 14 g cheese. This diffusion corresponds to a simultaneous perturbation along the patty and cheese axis. The signal attenuation factor μ shrinks the hamburger to the origin, the cheeseburger, in analogy to an exponential damping. The noise variance σ controls the magnitude of the injected noise and grows monotonically with time. Here we applied a large noise realization of $\boldsymbol{\epsilon} = [\,0, \sqrt{2}, 1\,]$ in a single step. Typically $\boldsymbol{\epsilon}$ is much smaller, and it would be highly unlikely to reach the Mc Double within a small number of steps.

Geometric interpretation *Geometrically, each burger is a point in $\mathbb{R}^3$, with coordinates given by its ingredient weights. Forward diffusion maps each point to a Gaussian cloud centered at $\mu(t)\boldsymbol{w}_0$. As the mean $\mu(t)$ decreases, the center contracts towards the origin. As the variance*

$\sigma(t)$ increases, the cloud expands towards a sphere. As time grows, distinct burgers become less distinguishable, because their Gaussian clouds overlap more strongly, and the distribution smoothens out. In the long-time limit, the influence of the initial data vanishes and the distribution approaches an isotropic spherical Gaussian distribution centered at the origin, that represents maximum uncertainty about the ingredient weights. This is the continuous analogue of the discrete random walk on the cube: Instead of probability mass spreading discretely between vertices, probability density spreads continuously in $\mathbb{R}^3$ and loses the sharp structure of the training data.

Analogy to mechanics. *We can describe the forward diffusion process by a Fokker–Planck equation, $\partial p/\partial t = -\nabla\cdot(\boldsymbol{b}\,p) + \frac{1}{2}\nabla\cdot(\boldsymbol{D}\cdot\nabla p)$, where $p(\boldsymbol{w},t)$ is the probability density of burgers with ingredient weights $\boldsymbol{w}$ at time t, $\boldsymbol{b} = -\frac{1}{2}\beta\,\boldsymbol{w}$ is the drift, and $\boldsymbol{D} = \beta\,\boldsymbol{I}$ is the diffusion tensor. For a constant diffusion tensor $\boldsymbol{D} = \beta\,\boldsymbol{I}$, this reduces to a drift–diffusion equation, $\partial p/\partial t = \frac{1}{2}\beta\nabla^2 p - \nabla\cdot(\frac{1}{2}\beta\,\boldsymbol{w}\,p)$, that highlights the combined effects of isotropic diffusion and a linear restoring force. This structure is directly analogous to transport equations in continuum mechanics, where advection and diffusion jointly govern the evolution of conserved quantities. Continuous forward diffusion takes the interpretation of the continuum limit of the discrete Markov process, in which the generator of the random walk converges to a diffusion operator. In this sense, forward diffusion provides a bridge between stochastic hopping on a finite state space and transport in a continuous ingredient-weight space, analogous to the transition from lattice models to continuum models in classical mechanics.*

3.2. Reverse diffusion

While forward diffusion gradually destroys structure by repeatedly noising the training data, reverse diffusion aims to reconstruct structure by progressively sharpening noisy samples. Instead of pushing burger points towards the origin and injecting noise, the reverse process defines a drift, which in general must be learned from data, that pulls noisy weight vectors back towards regions of high data density. Specifically, we introduce continuous reverse diffusion through a time-reversed stochastic differential equation that incorporates a *score function*,

$$\mathrm{d}\boldsymbol{w}_t = [\,g^2(t)\nabla_{\boldsymbol{w}}\log(p_t(\boldsymbol{w}_t)) - \boldsymbol{f}(\boldsymbol{w}_t,t)\,]\mathrm{d}t + g(t)\mathrm{d}\tilde{\boldsymbol{B}}_t\,, \tag{29}$$

where $\nabla_{\boldsymbol{w}}\log(p_t(\boldsymbol{w}))$ is the score function in terms of $p_t(\boldsymbol{w})$, the marginal distribution of $\boldsymbol{w}_t$, and $\mathrm{d}\tilde{\boldsymbol{B}}_t$ is reverse-time vector-valued stochastic differential of Brownian motion.

Ornstein–Uhlenbeck reverse process. Here, in analogy with the forward diffusion process, for reverse diffusion, we use the *time-reverse Ornstein–Uhlenbeck process* [44],

$$\mathrm{d}\boldsymbol{w}_t = [\,\tfrac{1}{2}\,\beta(t)\,\boldsymbol{w}_t + \beta(t)\nabla_{\boldsymbol{w}}\log(p_t(\boldsymbol{w}_t))\,]\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\tilde{\boldsymbol{B}}_t. \tag{30}$$

The first term, $\frac{1}{2}\beta(t)\boldsymbol{w}_t$, is the *reverse drift* that counteracts the contraction of the forward process and re-expands the state away from the origin. The second term, $\beta(t)\nabla_{\boldsymbol{w}}\log(p_t(\boldsymbol{w}_t))$, is the *score term* that steers samples along the direction of increasing likelihood towards the data manifold. Together, these two deterministic components define an effective probability force that reconstructs structure from noise. The third term, $\sqrt{\beta(t)}\,\mathrm{d}\tilde{\boldsymbol{B}}_t$, is the *stochastic diffusion* that injects controlled randomness to ensure sufficient exploration of the state space and prevent collapse onto a single trajectory. We can interpret reverse diffusion as a guided stochastic relaxation process that converts Gaussian noise into structured samples by following the gradient flow of the log-density while maintaining thermal fluctuations.

Score function. Our example permits an analytic evaluation of the score function, $\nabla_{\boldsymbol{w}}\log(p_t(\boldsymbol{w}))$, because the linear Gaussian forward process transforms the discrete training distribution into a tractable Gaussian mixture, for which the log-density gradient admits a closed-form expression. For a training set of N samples $\{\boldsymbol{w}_i\}_{i=1}^N$, the empirical data distribution is $p_{\text{data}}(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^N \delta(\boldsymbol{w}-\boldsymbol{w}_i)$. Under forward Ornstein–Uhlenbeck diffusion, each point mass evolves into a Gaussian with mean $\mu(t)\boldsymbol{w}_i$ and covariance $\sigma(t)\boldsymbol{I}$, and the forward marginal distribution at time t is available in closed form,

$$p_t(\boldsymbol{w}) = \frac{1}{N}\sum_{i=1}^N \mathcal{N}\left(\boldsymbol{w};\mu(t)\boldsymbol{w}_i,\sigma(t)\boldsymbol{I}\right), \tag{31}$$

with Ornstein–Uhlenbeck moments $\mu(t) = \exp(-\frac{1}{2}\alpha(t))$ and $\sigma(t) = 1-\exp(-\alpha(t))$. For the present example, the training set contains two burgers only, the cheeseburger with $\boldsymbol{w}_1 = [\,0,0,0\,]$ and the hamburger with $\boldsymbol{w}_2 = [\,0,0,-1\,]$, and this expression reduces to $p_t(\boldsymbol{w}) = \frac{1}{2}\mathcal{N}\left(\boldsymbol{w};\mu(t)\boldsymbol{w}_1,\sigma(t)\boldsymbol{I}\right) + \frac{1}{2}\mathcal{N}\left(\boldsymbol{w};\mu(t)\boldsymbol{w}_2,\sigma(t)\boldsymbol{I}\right)$. Taking the gradient of the log-density yields

$$\nabla_{\boldsymbol{w}}\log(p_t(\boldsymbol{w})) = \sum_{i=1}^N \gamma_i(\boldsymbol{w},t)\,\frac{\mu(t)\boldsymbol{w}_i - \boldsymbol{w}}{\sigma(t)} \quad\text{with}\quad \gamma_i(\boldsymbol{w},t) = \frac{\exp\left(-\|\boldsymbol{w}-\mu(t)\boldsymbol{w}_i\|^2/(2\sigma(t))\right)}{\sum_{j=1}^N \exp\left(-\|\boldsymbol{w}-\mu(t)\boldsymbol{w}_j\|^2/(2\sigma(t))\right)}\,, \tag{32}$$

where $\gamma_i(\boldsymbol{w},t)$ denotes the posterior probability weight of component i, that is, the probability that $\boldsymbol{w}$ originated from the ith training sample under the forward process. This expression shows that the score is a weighted average of linear restoring directions pointing from $\boldsymbol{w}$ towards the forward-diffused training samples $\mu(t)\boldsymbol{w}_i$. The sum in the denominator runs over the training data N. For the three-ingredient benchmark, we can evaluate the score function *exactly*, because the discrete training data has only two configurations, cheeseburger and hamburger. For the full 146-ingredient problem in Section 4, this exact computation becomes intractable, because the corresponding normalization sum would run over 2260 training configurations. In this case, we will evaluate the score function *approximately* by learning a model $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{w},t)$, for example a neural network with trainable parameters $\boldsymbol{\theta}$.

Numerical reverse sampling. To generate reverse trajectories, we discretize the reverse process with the *Euler–Maruyama* method [45]. At each reverse step k, we evaluate the score at the current state and at the corresponding forward time $t = T - k\Delta t$, which

decreases from T to 0 during sampling. The numerical update is

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + [\, \tfrac{1}{2}\, \beta \boldsymbol{w}_k + \beta\, \nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w}_k))\,]\Delta t + \sqrt{\beta \Delta t}\, \boldsymbol{\xi}_k \quad \text{with} \quad \boldsymbol{\xi}_k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}) \tag{33}$$

where $\boldsymbol{\xi}_k$ is a standard Gaussian random vector. This procedure transports samples from the noisy terminal mixture $p_T(\boldsymbol{w})$ back towards the empirical two-burger distribution at $t = 0$. This continuous reverse process is the analogue the discrete reverse process in Section 2, where we sample from the discrete posterior transition probabilities, for which Bayes' rule defines the reverse kernel exactly.

Analogy to mechanics. *We can interpret the reverse diffusion process as a stochastic gradient flow that reconstructs the data distribution, analogous to drift towards energy minima in dissipative mechanical systems. In the deterministic limit, this reduces to a gradient flow,* $\mathrm{d}\boldsymbol{w}/\mathrm{d}t = -\nabla U(\boldsymbol{w}, t)$, *with* $U(\boldsymbol{w}, t) = -\log(p_t(\boldsymbol{w}))$, *consistent with variational formulations of dissipative systems. The learned score function plays the role of an effective driving force that guides trajectories towards regions of high probability density. In this interpretation, the dynamics take the form,* $\mathrm{d}\boldsymbol{w} = -\nabla U(\boldsymbol{w})\,\mathrm{d}t + \sqrt{\beta}\,\mathrm{d}\boldsymbol{w}_t$, *with an effective potential,* $U(\boldsymbol{w}) = -\log(p(\boldsymbol{w}))$, *so that the score function,* $\nabla \log(p(\boldsymbol{w}))$, *corresponds to a conservative force driving the system towards equilibrium, while the additional linear term reflects the restoring drift inherited from the forward Ornstein–Uhlenbeck process.*

Relation to discrete diffusion. In the discrete diffusion model, the reverse process learns a conditional probability, $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$, by minimizing a cross-entropy loss. This loss is equivalent to minimizing the Kullback–Leibler divergence between the true and learned reverse transition kernels. In the continuous diffusion model, the reverse process learns the score function, $\nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w}))$, by minimizing a score-matching loss. This loss is equivalent to minimizing a time-integrated Kullback–Leibler divergence between the true reverse-time stochastic process and the model-implied reverse-time process. In both cases, forward diffusion increases entropy, reverse diffusion learns dynamics that reduce entropy, and training minimizes a divergence between true and learned reverse processes. The discrete cross-entropy loss and the continuous score-matching loss are two manifestations of the same principle: learning the time-reversed dynamics of diffusion.

Burger interpretation. *In the continuous burger example, the true reverse-time dynamics reflect the structure of the training weight distribution. Recall that, for our given training data, under* $p_{\text{data}}(\boldsymbol{w})$, *the bun weight is 55 g, the patty weight 45 g, and the cheese weight either 14 g or 0 g, with normalized coordinates* $\boldsymbol{w}_1 = [\,0, 0, 0\,]$ *and* $\boldsymbol{w}_2 = [\,0, 0, -1\,]$. *Suppose we observe a noisy burger,* $\boldsymbol{w}_t = [\,0, -1, 0\,]$, *a cheese sandwich with a negative patty coordinate,* $w^*_{\text{patty}} = -1$, *that translates into a zero patty weight,* $w_{\text{patty}} = 0\,\text{g}$. *Under the true reverse-time distribution, it is highly unlikely that a realistic previous burger has a zero patty weight. Therefore, the true score,* $\nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w}))$, *points in a direction that increases the patty weight and moves the sample towards the high-density region, near* $\boldsymbol{w}_t = [\,0, 0, 0\,]$ or $\boldsymbol{w}_t = [\,0, 0, -1\,]$. *If the learned score,* $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{w}_t, t)$, *predicts a different direction, the score-matching loss* $\mathcal{L}(\theta)$ *between the true reverse-time process and the learned reverse-time process increases. Minimizing the score-matching loss ensures that the reverse diffusion process restores realistic ingredient weights, while allowing natural continuous variability around the learned burger manifold.*

Geometric interpretation. *Forward diffusion corresponds to a probability flow that contracts samples towards the origin and adds isotropic Gaussian noise, to converge towards a spherical Gaussian distribution. Reverse diffusion introduces a learned drift term that follows the gradient field of the log-density, and steers samples towards regions of high probability under the data distribution. Geometrically, forward diffusion spreads the probability mass, while reverse diffusion pulls it back towards the data manifold.*

Analogy to mechanics. *From a mechanics perspective, the reverse process defines time-reversed stochastic dynamics, in which the score function,* $\nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w}))$, *acts as an effective force that drives the system towards high-probability configurations. This is analogous to gradient flow in an energy landscape, where the log-density plays the role of a free energy. The resulting dynamics resemble rare-event sampling and barrier-crossing processes such as Kramers escape problems* [46], *and connect to probabilistic path formulations of stochastic dynamics in the spirit of Onsager's principle* [47].

3.3. Continuous forward and reverse diffusion

We now illustrate the behavior of continuous diffusion in the three-dimensional ingredient space defined by bun, patty, and cheese weights (Figs. 7 and 8). In analogy to the discrete diffusion example (Figs. 4 and 5), we adopt a constant diffusion rate, $\beta(t) = \beta$, for which the integrated noise simplifies to $\alpha(t) = \beta t$, the signal attenuation factor is $\mu(t) = e^{-\beta t/2}$, and the noise variance is $\sigma(t) = (1 - e^{-\beta t})$, with a time T=1 and 100 steps. During *forward diffusion* (Fig. 7), the model begins with a distribution concentrated on two equally probable training burgers, cheeseburger and hamburger, whose ingredient weights define two distinct points in the continuous space. As diffusion proceeds, stochastic trajectories spread outward from these initial states, leading to increasing variability in bun, patty, and cheese weights (left). This spreading is reflected in the progressive broadening of the probability density in the cheese space (center), where initially sharp peaks evolve into a diffuse distribution. Over time, the distribution approaches a Gaussian-like equilibrium with maximal entropy, as evidenced by the convergence of the entropy curves towards a plateau (right). This behavior reflects the role of forward diffusion as a stochastic mixing process that continuously perturbs and ultimately erases information about the initial configuration (Fig. 9, left). During *reverse diffusion* (Fig. 8), the model inverts the diffusion dynamics. Rather than starting from unconstrained Gaussian noise, we initialize the reverse process by sampling directly from the terminal forward distribution $p_T(\boldsymbol{w})$. Reverse diffusion progressively contracts trajectories back towards the training states. This manifests itself in a convergence of ingredient weights towards the characteristic values of the cheeseburger and hamburger (left), accompanied by a re-localization of probability density into concentrated regions in the cheese space (center). The entropy decreases towards

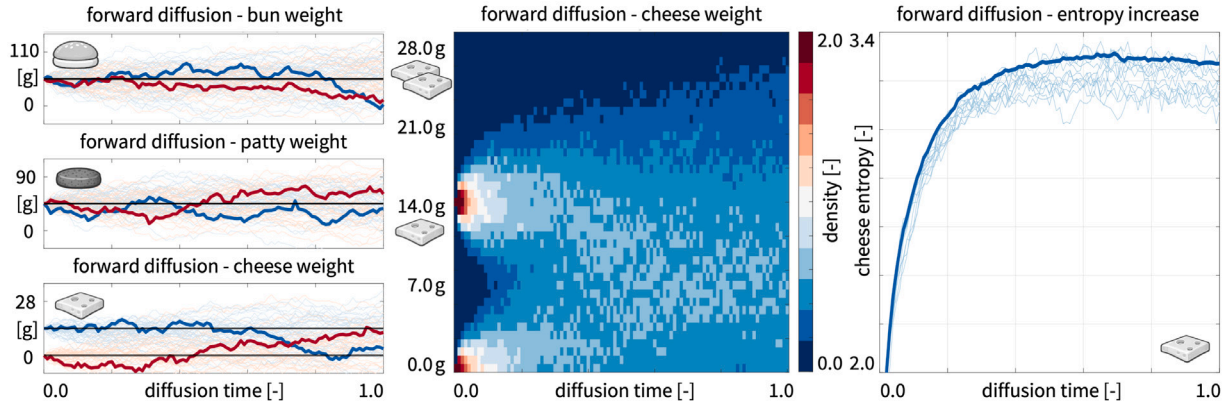

Fig. 7. Forward diffusion in continuous ingredient space. Forward diffusion creates a noised state from the training data. Starting from two training states, cheeseburger and hamburger, individual trajectories progressively diffuse and spread in the weight space as reflected by the evolution of ingredient weights (left), broadening of the cheese probability density (center), and increase in cheese entropy (right).

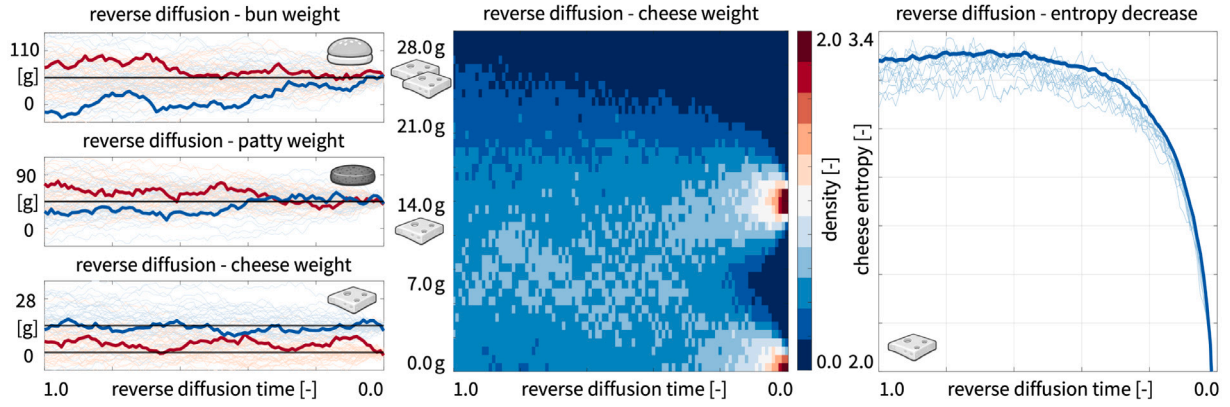

Fig. 8. Reverse diffusion in continuous ingredient space. Reverse diffusion reconstructs the training distribution from the noised state. Starting from a diffuse distribution, trajectories progressively concentrate back onto the original two burgers as reflected by the convergence of ingredient weights (left), re-localization of the probability density (center), and decrease in cheese entropy (right). Thin lines indicate stochastic realizations; thick lines show ensemble averages.

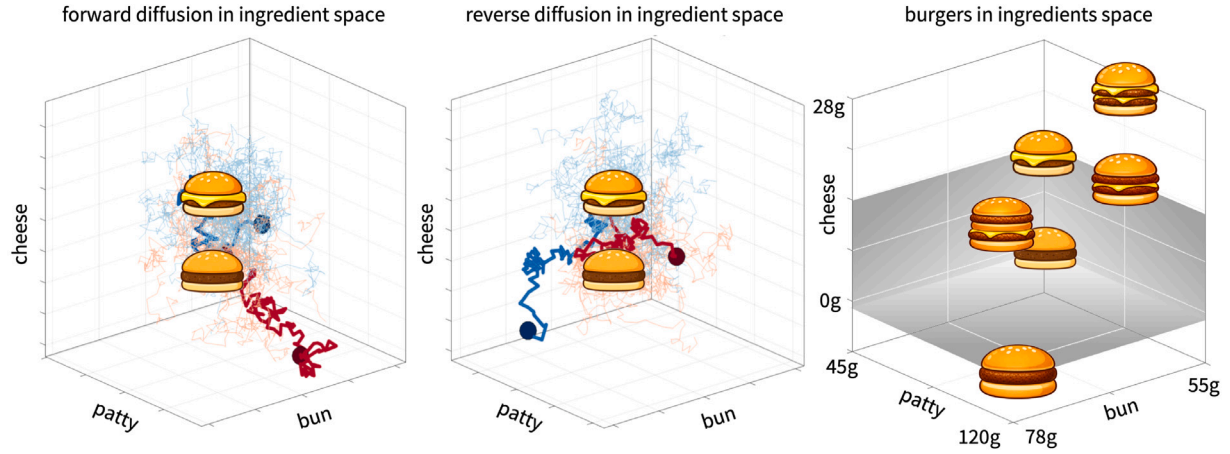

Fig. 9. Forward and reverse diffusion in continuous ingredient space. Forward diffusion creates the noised state from the training data, Hamburger (red lines) and Cheeseburger (blue lines) (left). Reverse diffusion reconstructs the training data, Hamburger (red lines) and Cheeseburger (blue lines), from the noised state (middle). Illustration of the training data, Hamburger and Cheeseburger, and other possible burgers, Mc Double, Big Mac, Double Cheeseburger, and Quarter Pounder, in ingredient weight space (right).

Table 1
Generating new burgers by continuous diffusion. Burgers, weights, coordinates, squared distance to training manifold, and probabilities p_{path} and p_{end} that a single trajectory passes through or ends at the target burger at a 20% tolerance for continuous forward diffusion with 5M sampled trajectories with 100 steps.

burger	burger geometry			diffusion rate $\beta = 0.10$		diffusion rate $\beta = 0.25$	
	$\boldsymbol{w}^*$	$\boldsymbol{w}$	d	p_{path}	p_{end}	p_{path}	p_{end}
Hamburger	[55, 45, 0] g	[0, 0,-1]	0.000	0.5012912	0.0562541	0.5125357	0.0194421
Cheeseburger	[55, 45, 14] g	[0, 0, 0]	0.000	0.5028274	0.0572939	0.5201618	0.0212541
Mc Double	[55, 90, 14] g	[0, 1, 0]	1.000	0.0016643	0.0005516	0.0146338	0.0025231
Big Mac	[78, 90, 14] g	[0.418, 1, 0]	1.084	0.0007236	0.0002504	0.0094108	0.0017217
Double Cheeseburger	[55, 90, 28] g	[0, 1, 1]	1.414	0.0000136	0.0000054	0.0011225	0.0002582
Quarter Pounder	[72, 120, 14] g	[0.309, 1.667, 0]	1.695	0.0000002	0.0000002	0.0001581	0.0000438

its initial value (right) as the burgers recover structured information (Fig. 9, middle). The thin lines of the individual stochastic trajectories visualize the variability, while the thick lines highlight the path of a single trajectory. Taken together, these results demonstrate that, in the continuous setting, diffusion operates over a continuous state space, where forward diffusion spreads probability mass into a high-dimensional Gaussian distribution and the learned reverse dynamics reconstruct the underlying data manifold.

3.4. Generating new burgers by continuous diffusion

We quantify how well continuous diffusion generates burgers beyond the two training examples, hamburger and cheeseburger. We define a burger as *discovered* when a sampled trajectory enters a tolerance box of 20% around the target burger, with dimensions of $\Delta\boldsymbol{w}^* = [\pm 11.0, \pm 9.0, \pm 2.8]$ g in bun, patty, and cheese. We estimate both the pathwise discovery probability p_{path}, which counts whether a trajectory visits the target at *any time*, and the endpoint discovery probability p_{end}, which counts whether the *end state* lands inside the tolerance box. We sample 5 million trajectories over a total time of $T = 1$ with 100 steps at $\Delta t = 0.01$ each. We report the results for six burgers in both raw weight space $\boldsymbol{w}^*$ and transformed cheeseburger-centered coordinates $\boldsymbol{w}$ (Table 1). The training data define a one-dimensional manifold, the line segment connecting hamburger and cheeseburger. The burgers we seek to discover are located progressively further away from this manifold: McDouble at a distance d=1, Big Mac at d=1.084, Double Cheeseburger at d=1.414, and Quarter Pounder at d=1.695. The probabilities p_{path} and p_{end} quantify the likelihood that a single trajectory passes through or ends at a defined target burger. Discovery difficulty grows sharply with distance from the training manifold (Fig. 10). Specifically, $N_{95} = [\log(0.05)/\log(1-p)]$ denotes the number of independent samples required to achieve 95% probability of discovery. Its logarithm, $\log_{10}(N_{95})$, is approximately linear in the distance squared, d^2, which implies an approximately exponential increase in the number of required samples with squared geometric distance from the training data. For a diffusion rate of β=0.10, the slopes of this linear relation, visualized through the dashed and solid lines are 1.928 and 2.263 for endpoint and pathwise discovery (Fig. 10, left). Increasing the diffusion rate to β=0.25 notably decreases these slopes to 0.935 and 1.266 (Fig. 10, right), indicating a broader exploration of the ingredient-weight space under stronger diffusion.

Analogy to mechanics. *The observed exponential growth of sampling cost with squared distance is analogous to rare-event dynamics in stochastic systems, where transition probabilities decay exponentially with an effective energy barrier. In this interpretation, the squared distance from the training manifold plays the role of a barrier height, and discovering new burgers corresponds to a first-passage event across this barrier. This behavior is consistent with Arrhenius-type scaling* [48] *in which transition rates depend exponentially on barrier height,* $N_{95} \sim \exp(c\, d^2)$, *where* c *is a constant that depends on the diffusion rate* β.

The individual burgers follow this trend: At small diffusion, β=0.10, the Mc Double, which differs from cheeseburger only through a second patty, requires N_{95}^{path}=1799 and N_{95}^{end}=5430 samples for pathwise and endpoint discovery. The Big Mac, which combines an increased bun weight with a second patty, is slightly farther from the training manifold and requires 4139 and 11,963 samples. The Double Cheeseburger, which moves simultaneously in patty and cheese, is markedly harder to discover, with 220,273 and 554,764 samples. The Quarter Pounder is furthest from the training manifold and becomes effectively unreachable at β=0.10, as it requires approximately 1.5×10^7 samples for pathwise and endpoint discovery. Increasing the diffusion rate to β=0.25 substantially lowers the discovery cost of unseen burgers: The Mc Double drops from N_{95}^{path}=1799 to 204, the Big Mac from 4139 to 317, the Double Cheeseburger from 220,273 to 2675, and the Quarter Pounder from 1.5×10^7 to 18,959 for pathwise discovery. Endpoint discovery remains consistently more expensive than pathwise discovery, since it requires the *final* state rather than *any* intermediate state to reach the target. At the same time, stronger diffusion reduces endpoint retention of the training burgers themselves, and increases the N_{95}^{end} of the hamburger and cheeseburger from 50 to 150. This reveals a trade-off: Increasing the diffusion rate β improves the exploration of unseen burgers, but reduces the probability of ending near the original training modes. Taken together, this example shows that continuous diffusion does not generate new burgers uniformly. Instead, it preferentially explores a neighborhood around the training manifold, and the cost of discovering new burgers grows rapidly with distance in transformed ingredient-weight space.

Analogy to mechanics. *Increasing the diffusion rate* β *reduces the characteristic relaxation time and allows trajectories to explore larger regions of the state space within a fixed time horizon. This is analogous to increasing the temperature in Langevin dynamics* [49], *where stronger stochastic forcing accelerates relaxation and broadens the distribution around equilibrium. In both cases, higher noise levels enhance exploration while reducing retention near equilibrium configurations.*

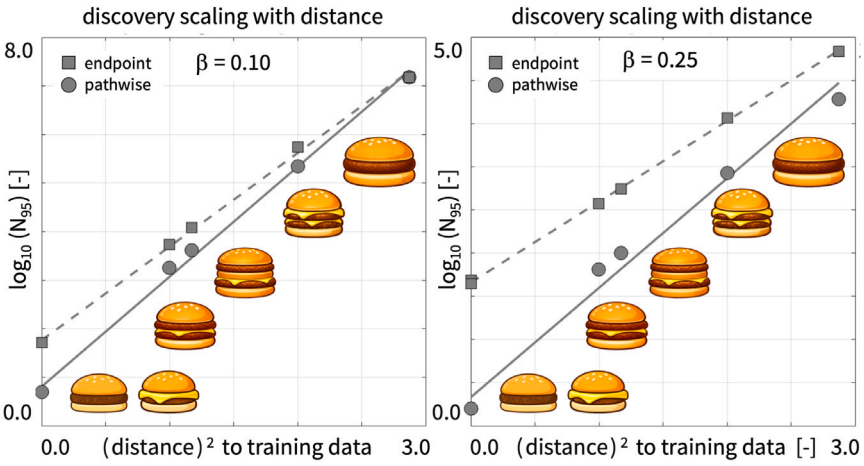

Fig. 10. Generating new burgers by continuous diffusion. Sampling complexity for discovering new burgers, Mc Double, Big Mac, Double Cheeseburger, and Quarter Pounder, beyond the training data, Hamburger and Cheeseburger. Number of sample trajectories required for 95% discovery vs. squared distance to training manifold for diffusion rates β=0.10 (left) and β=0.25 (right). Squares and dashed lines indicate endpoint discovery; circles and solid lines indicate pathwise discovery for paths with 100 steps. Discovery scales linearly with the distance squared, on the logarithmic scale, implying an exponential growth of discovery cost with squared distance from training data.

4. Generative AI for burgers

We now extend the discrete and continuous diffusion formulations from the three-ingredient benchmark with $n = 3$ ingredients, two training burgers, and $2^3 = 8$ possible burgers to a real-world problem with $n = 146$ ingredients and 2260 training burgers. Similar to Sections 2 and 3, we represent each burger through a binary ingredient mask $\boldsymbol{x}$ that indicates the presence or absence of each ingredient and the ingredient weight $\boldsymbol{w}$ that specifies the weight in grams,

$$\boldsymbol{x} \in \{0,1\}^n \quad \text{and} \quad \boldsymbol{w} \in \mathbb{R}^n . \tag{34}$$

Notably, the number of possible burgers grows exponentially with the number of ingredients as 2^n, which renders exhaustive exploration infeasible and motivates generative modeling. For 146 independent ingredients that can be present or absent, the discrete design space consists of $2^{146} = 8.92\times10^{43}$ possible burgers. To generate candidate burgers, we integrate a *discrete diffusion model* [50] that generates ingredient masks with a *continuous diffusion model* [13] that generates ingredient weights, conditional on a given mask. Yet, the transition from the three-ingredient benchmark to the real-world problem marks a fundamental shift in the nature of the problem: in the *low-dimensional setting* of Sections 2 and 3, the reverse dynamics are analytically tractable, the score function is known, the reverse transitions are computable, and the underlying physics admit a closed-form description. In contrast, in the *high-dimensional setting*, the score function becomes intractable, the reverse dynamics can no longer be computed explicitly, and the physics must be inferred from data through learned models.

4.1. Discrete diffusion

We model ingredient selection using a multinomial diffusion process [50], that directly extends the discrete diffusion formulation introduced in Section 2 to a high-dimensional setting. We represent ingredient presence through a binary vector, $\boldsymbol{x} \in \{0,1\}^n$, which we modulate via forward and reverse diffusion. The *forward diffusion* process gradually destroys structure as

$$p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \mathcal{C}\big(\boldsymbol{x}_t \,\big|\, [\,1-\beta_t\,]\boldsymbol{x}_{t-1} + \beta_t/K\big)\,, \tag{35}$$

where $\mathcal{C}$ denotes a *categorical distribution* over K categories applied independently to each ingredient. Its parameter is $([\,1-\beta_t\,]\boldsymbol{x}_{t-1} + \beta_t/K)$, where β_t is the flip probability at time t, and K is the number of categories. In our application, ingredient selection is *binary*, $K = 2$, meaning an ingredient is either present or absent. We can reparameterize Eq. (35) to obtain a Bernoulli distribution with parameter $([\,1-\beta_t\,]\boldsymbol{x}_{t-1} + \beta_t\,[\,1-\boldsymbol{x}_{t-1}\,])$, which flips the ingredient state with a probability β_t and keeps it the same with a probability β_t. Since these distributions form a Markov chain with independent Bernoulli corruption,

$$p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}) = \prod_{i=1}^{n} p(x_{t,i} \mid x_{t-1,i}) \quad \text{with} \quad p(x_{t,i} = 1 \mid x_{t-1,i}) = [\,1-\beta_t\,]x_{t-1,i} + \tfrac{1}{2}\,\beta_t\,, \tag{36}$$

we can explicitly calculate the distribution at any time t in terms of the initial data $\boldsymbol{x}_0$ and the cumulative retention factor $\bar{\alpha}_t$,

$$p(\boldsymbol{x}_t \mid \boldsymbol{x}_0) = \mathcal{C}(\boldsymbol{x}_t \mid \bar{\alpha}_t\,\boldsymbol{x}_0 + \tfrac{1}{2}\,[\,1-\bar{\alpha}_t\,]) \quad \text{with} \quad \bar{\alpha}_t = \prod_{\tau=1}^{t}[\,1-\beta_\tau\,]\,, \tag{37}$$

where $\bar{\alpha}_t$ is the product of the per-step retention factors $[\,1-\beta_\tau\,]$, which quantifies how much of the original ingredient configuration $\boldsymbol{x}_0$ remains after t diffusion steps.

The *reverse diffusion* process reconstructs structure by approximating the time-reversed transition probabilities. During training, the clean configuration $\boldsymbol{x}_0$ is known, which allows us to evaluate the posterior distribution via Bayes' theorem,

$$p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})\, p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)}{\sum_{\boldsymbol{y}\in\{0,1\}^n} p(\boldsymbol{x}_t \mid \boldsymbol{y})\, p(\boldsymbol{y} \mid \boldsymbol{x}_0)}. \tag{38}$$

The *likelihood* $p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$ denotes the forward diffusion transition probability, which is given by independent Bernoulli flips with probability β_t for each ingredient. The *prior* $p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)$ represents the marginal distribution of the forward process at time $(t-1)$, obtained from the closed-form solution of the diffusion process. The *evidence* acts as a normalization constant that ensures the posterior sums to one. It is the sum over all possible configurations $\boldsymbol{y} \in \{0,1\}^n$ of the product $p(\boldsymbol{x}_t \mid \boldsymbol{y})\, p(\boldsymbol{y} \mid \boldsymbol{x}_0)$, which corresponds to the total probability of observing $\boldsymbol{x}_t$ under all possible previous states. As such, this *posterior* combines likelihood information from the noisy state $\boldsymbol{x}_t$ with prior information propagated from the original configuration $\boldsymbol{x}_0$. Because the forward process factorizes across ingredients, this update operates independently for each ingredient. For each ingredient i, we obtain a Bernoulli distribution,

$$p(x_{t-1,i} \mid x_{t,i}, x_{0,i}) \propto p(x_{t,i} \mid x_{t-1,i})\, p(x_{t-1,i} \mid x_{0,i}), \tag{39}$$

with normalization over $x_{t-1,i} \in \{0,1\}$. While this posterior provides a closed-form expression for the reverse dynamics during training, its evaluation requires knowledge of $\boldsymbol{x}_0$ and is therefore not directly usable at generation time. We therefore introduce a neural network model $\mu_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$ where $\boldsymbol{\theta}$ denotes the network parameters to predict the clean configuration from a noisy sample [12],

$$\hat{\boldsymbol{x}}_0 = \mu_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \approx \mathbb{E}[\,\boldsymbol{x}_0 \mid \boldsymbol{x}_t\,] \tag{40}$$

We use this estimate to approximate the intractable posterior by replacing the unknown $\boldsymbol{x}_0$,

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t) \approx p(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \hat{\boldsymbol{x}}_0). \tag{41}$$

The discrete diffusion model defines the ingredient selection $\boldsymbol{x}$, which conditions the continuous diffusion model to generate the corresponding ingredient weights $\boldsymbol{w}$.

Geometric interpretation. *The discrete diffusion process defines a random walk on a* 146*-dimensional hypercube, where vertices represent ingredient configurations and edges correspond to single ingredient flips. Forward diffusion spreads probability mass locally and progressively disperses it across the hypercube. In high dimensions, however, concentration effects dominate: most configurations lie near the boundary, typical distances increase, and the distribution rapidly approaches a high-entropy regime in which structure is difficult to distinguish. Reverse diffusion must therefore recover structure from weak signals in a space where relevant configurations occupy only a small fraction of the domain. This highlights the need for learned generative models that capture correlations between ingredients and guide probability mass towards realistic configurations.*

4.2. Continuous diffusion

We model ingredient quantification using a score-based diffusion process in which we represent ingredient weights similar to Section 3 as a vector of real-valued variables, $\boldsymbol{w} \in \mathbb{R}^n$, that we modulate via forward and reverse diffusion. In analogy with Section 3, we model *forward diffusion* using the Ornstein–Uhlenbeck process [43],

$$\mathrm{d}\boldsymbol{w}_t = -\tfrac{1}{2}\,\beta(t)\,\boldsymbol{w}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{B}_t\,, \tag{42}$$

and *reverse diffusion* using the time-reverse Ornstein–Uhlenbeck process [13],

$$\mathrm{d}\boldsymbol{w}_t = [\,\tfrac{1}{2}\,\beta(t)\,\boldsymbol{w}_t + \beta(t)\nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w}_t))\,]\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\tilde{\boldsymbol{B}}_t, \tag{43}$$

where $\mathrm{d}\boldsymbol{w}_t$ denotes the stochastic differential that represents the infinitesimal random increment of the process. Both equations share *deterministic drift*, forward $-\frac{1}{2}\beta(t)\boldsymbol{w}_t$ or reverse $+\frac{1}{2}\beta(t)\boldsymbol{w}_t$, and *stochastic diffusion*, $\sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{B}_t$ or $\sqrt{\beta(t)}\,\mathrm{d}\tilde{\boldsymbol{B}}_t$ in terms of the stochastic differential of the Brownian motion $\mathrm{d}\boldsymbol{B}_t$ or $\mathrm{d}\tilde{\boldsymbol{B}}_t$. In addition, reverse diffusion also contains the *score function*, $\nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w}))$, a vector field that drives diffusion uphill towards higher probability regions [51,52]. The time-varying diffusion rate $\beta(t) > 0$ controls the magnitude of both drift and diffusion. It increases as time progresses, $\beta(t) = \beta_{\min} + t\,[\,\beta_{\max} - \beta_{\min}]$, from an initial weak contraction and weak noise at $\beta_{\min}$ towards a strong contraction and strong noise at $\beta_{\max}$. Similar to Section 3, the stochastic differential Eq. (42) has the following closed-form conditional distribution,

$$p(\boldsymbol{w}_t \mid \boldsymbol{w}_0) = \mathcal{N}(\,\boldsymbol{w}_t \mid \mu(t)\,\boldsymbol{w}_0, \sigma(t)\boldsymbol{I}\,)\,, \tag{44}$$

with

$$\mu(t) = \exp(-\tfrac{1}{2}\alpha(t)\,) \quad \text{and} \quad \sigma(t) = 1 - \exp(-\alpha(t)) \quad \text{and} \quad \alpha(t) = \textstyle\int_0^t \beta(s)\,\mathrm{d}s\,, \tag{45}$$

where $\mu(t)$ is the signal attenuation factor $\sigma(t)$ is the noise variance accumulated by time t, and $\alpha(t)$ is the integrated noise.

Score function. The score function, $\nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w}))$, is the gradient of the log-density at time t. It points in the direction of the steepest increase in probability density. In contrast the low-dimensional system in Section 3, where we could evaluate the score function

explicitly, the high-dimensional setting makes the marginal density $p_t(\boldsymbol{w})$ intractable: Analytically evaluating the score function would require integration over all training configurations, which becomes computationally prohibitive. We therefore approximate the vector-valued score function by a neural network model [13],

$$\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{w}, t) \approx \nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w}))\,, \tag{46}$$

parametrized in terms of the parameters $\boldsymbol{\theta}$, the network weights and biases. We train the neural networks to learn the score $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{w}, t)$ by minimizing the *score-matching* loss $\mathcal{L}$, the error between the true score function, $\nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w}))$, and the approximated score of the model, $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{w}, t)$,

$$\mathcal{L}(\boldsymbol{\theta}) = \int_0^T \mathbb{E}_{p_t}[\, \|\nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w})) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{w}, t)\|^2\,]\mathrm{d}t\,, \tag{47}$$

where $\|\cdot\|$ is the Euclidean norm, $\mathbb{E}_{p_t}$ denotes the expectation taken over noisy samples $\boldsymbol{w}_t$, and the integral over t averages the loss across all noise levels. The term inside the expectation, $\|\nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w})) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{w}, t)\|^2$, penalizes discrepancies between the true direction in which the probability density increases most, $\nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w}))$, and the direction predicted by the model, $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{w}, t)$. By minimizing the loss function over the parameter space $\boldsymbol{\theta}$, as $\boldsymbol{\theta}^* = \min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$, we find the model parameters $\boldsymbol{\theta}^*$ that best approximate the true gradient field of the log-density.

Continuous diffusion conditioned on ingredient selection. Eqs. (42) to (49) present continuous diffusion modeling in its general form. In practice, however, we seek to model a *conditional probability distribution*, $p(\boldsymbol{w}|\boldsymbol{x})$, conditioned on the recipe mask $\boldsymbol{x}$ produced by the discrete diffusion model. We therefore model the conditional distribution $p_t(\boldsymbol{w} \mid \boldsymbol{x})$, which we induce by conditioning the initial data distribution $p(\boldsymbol{w}_0 \mid \boldsymbol{x})$ and propagating it through the forward diffusion process. This conditioning allows the model to generate ingredient weights that are consistent with a given ingredient selection while preserving the statistical structure of the training data. Accordingly, we redefine the score function as

$$\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{w}, t) \approx \nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w}|\boldsymbol{x}))\,. \tag{48}$$

Now, instead of training the score-matching loss (49) to learn the score function $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{w}, t)$, we train the revised score-matching loss,

$$\mathcal{L}(\boldsymbol{\theta}) = \int_0^T \mathbb{E}_{p_t}[\, \|\nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w} \mid \boldsymbol{x})) - \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{w}, t)\|^2\,]\mathrm{d}t\,, \tag{49}$$

to learn the revised score function $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{w}, t)$ to approximate the gradient field of the conditional log-density, $p_t(\boldsymbol{w}|\boldsymbol{x})$. Combined with the discrete diffusion model for ingredient selection, this conditional formulation enables the generation of complete burgers defined by both ingredient selection $\boldsymbol{x}$ and ingredient weights $\boldsymbol{w}$.

Training and validation. We construct the *mask model* using a neural network with an embeddings layer with 1000 embeddings for the time variable, and three fully connected layers of 512 neurons each to predict the logits of the mask model. We then use the logits with the softmax function to produce the raw probabilities. We use minibatching with $n = 1000$ for training and train with a learning rate of $5 \cdot 10^{-4}$ using the Adam optimizer for 100,000 epochs. We construct the *value model* using a feed-forward neural network with four hidden layers of 256 neurons each. The vectors for $\boldsymbol{x}$ and $\boldsymbol{w}$ form the input for s_{θ}. We use a batch size of 400 and a learning rate of $1 \cdot 10^{-3}$ with the Adam optimizer and train for 20,000 epochs. We split the training, 80% for training and 20% for validation and use Nvidia H100 and L40S GPUs. We use 1000 steps for the noising and denoising processes and select scaling parameters $\beta_{\min} = 0.001$ and $\beta_{\max} = 3$ [53].

4.3. Discrete and continuous diffusion in high-dimensional ingredient space

We now illustrate the behavior of discrete and continuous diffusion in a high-dimensional ingredient space defined by $n = 146$ ingredients (Fig. 11). In contrast to the three-ingredient benchmark with only $2^3 = 8$ possible burgers and two training samples, the present setting spans an exponentially large combinatorial design space of $2^{146} = 8.92 \times 10^{43}$ possible ingredient combinations, while the available training data comprise only 2260 burgers. This *extreme sparsity* fundamentally changes the nature of the problem: we can no longer solve the reverse dynamics analytically; instead, we must learn them from data. Fig. 11 demonstrates that, despite this challenge, diffusion retains its characteristic behavior. Forward diffusion progressively destroys structure by randomizing ingredient presence and weights, and drives the system towards a high-entropy state. Reverse diffusion – learned through neural networks – reconstructs structured burgers by concentrating probability mass onto the data manifold defined by the training set. The resulting stochastic trajectories exhibit variability at the individual level, while maintaining a consistent ensemble behavior that indicates that the learned reverse dynamics successfully capture the underlying statistical structure of the high-dimensional ingredient distribution. Taken together, these results show that diffusion models extend naturally from low-dimensional analytical settings to realistic, high-dimensional design spaces, where they act as data-driven operators that transform noise into structured, physically meaningful configurations.

4.4. Generating new burgers in high-dimensional space

We next evaluate the ability of the learned diffusion model to generate new burgers beyond the 2260 training examples in a space of 2^{146} possible designs. We compare the statistical properties of 1,000,000 generated burgers with the training data for both discrete diffusion (Fig. 12) and continuous diffusion (Figs. 13 and 14). The close agreement in ingredient counts, marginal

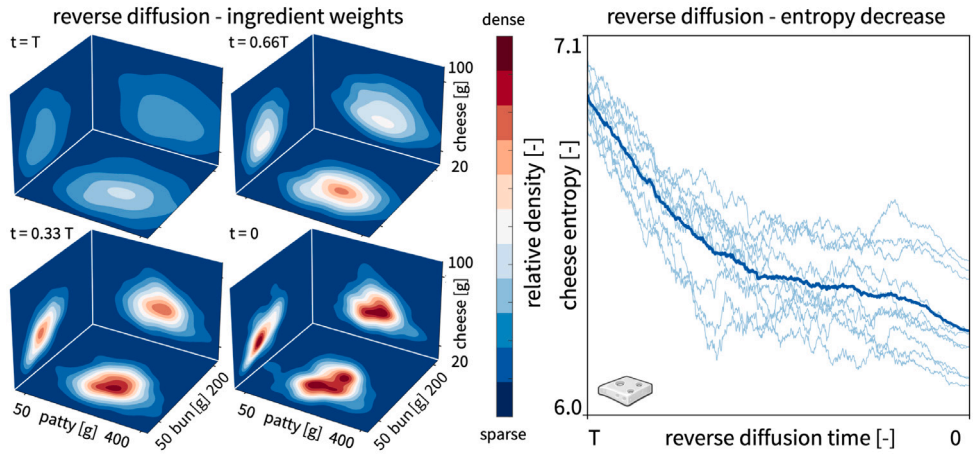

Fig. 11. Reverse diffusion in high-dimensional continuous ingredient space. Reverse diffusion recovers the training distribution from the uniform state. Starting from the noised state, probability mass concentrates back onto the original two burgers as reflected by the re-localization in probability space (left), and the entropy decrease (right). Thin lines indicate stochastic realizations; thick lines show ensemble averages.

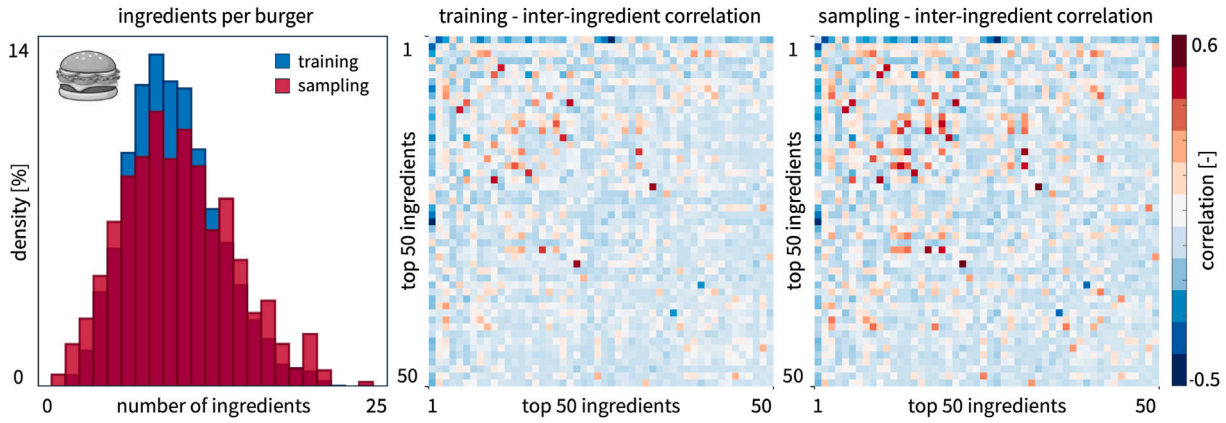

Fig. 12. Discrete diffusion in high-dimensional space. Comparison of 2260 training burgers and 1,000,000 samples generated by the learned discrete diffusion model. Distribution of the number of ingredients per burger (left); inter-ingredient correlation matrix for the top 50 most frequent ingredients in the training set (middle) and in the generated samples (right). The close agreement between training and sampling statistics demonstrates that the model accurately captures both marginal distributions and pairwise correlations in the discrete ingredient space.

probabilities, pairwise correlations, rare ingredients, and weight distributions demonstrates that the model accurately preserves both low-order statistics and higher-order dependencies of the data. This includes not only the most frequent ingredients, but also low-frequency events and the full correlation structure across all ingredients. This indicates that the learned generative process faithfully captures the underlying data distribution, even in an extremely high-dimensional and sparsely sampled regime. At the same time, the generated burgers are not simple reproductions of the training set. Instead, the model samples novel combinations of ingredients and weights that lie off the observed data manifold while remaining statistically consistent with it. This is the key hallmark of generative modeling: the ability to *interpolate*, *extrapolate*, and *explore* unseen regions of a vast combinatorial design space.

From the generated samples, we select five AI-generated burgers for experimental validation in a sensory study, using a previously reported dataset [26]. While the previous study focuses on evaluating AI-generated burgers in terms of sensory performance, nutrition, and environmental impact, here, we use a subset of these results solely to illustrate the proposed diffusion-based framework. We do not explicitly optimize or design these burgers in the classical sense, but sample them from the learned generative distribution, which is asymptotically dense in the high-dimensional design space and provides representative realizations of feasible configurations. We prepare the burgers according to the AI-generated ingredient lists, and evaluate them in a blinded sensory study with n = 101 participants in a real restaurant setting (Fig. 15). Remarkably, three of the AI-generated burgers outperform the classical Big Mac in overall liking, the delicious burger 2 significantly (5.7 ± 1.2 vs. 5.3 ± 1.5, $p<0.05$), while the delicious burger 1 (5.5 ± 1.4) and sustainable burger 2 (5.4 ± 1.6) show higher mean scores without reaching statistical significance. In flavor, both delicious burgers significantly outperform the Big Mac (both 5.8 ± 1.3 vs. 5.4 ± 1.5, $p<0.05$). Notably, the sustainable burger 2,

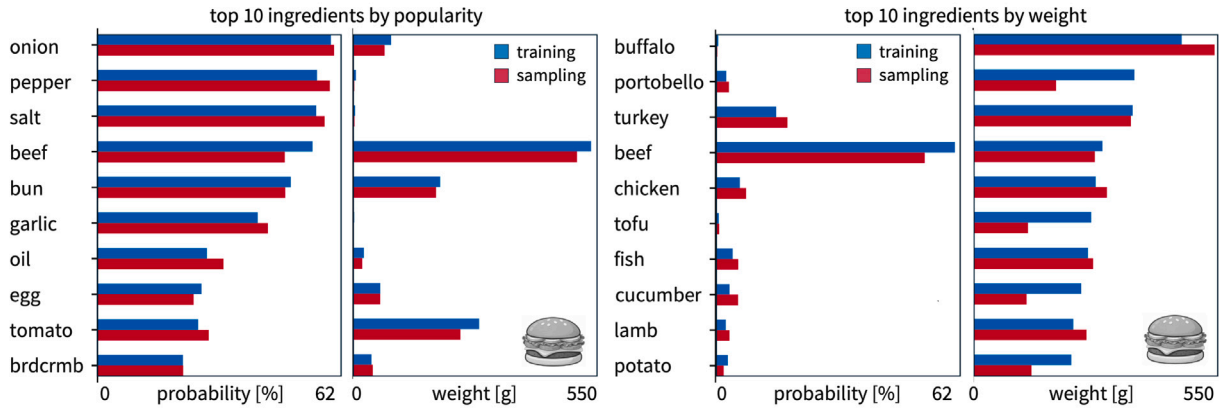

Fig. 13. Continuous diffusion in high-dimensional space. Comparison of 2260 training burgers and 1,000,000 samples generated by the learned continuous diffusion model. Occurrence probability of the top 10 most frequent ingredients (left) and average ingredient weights for the most frequent ingredients by total weight (right). The close agreement between training and sampling statistics demonstrates that the model accurately captures both ingredient prevalence and quantitative composition.

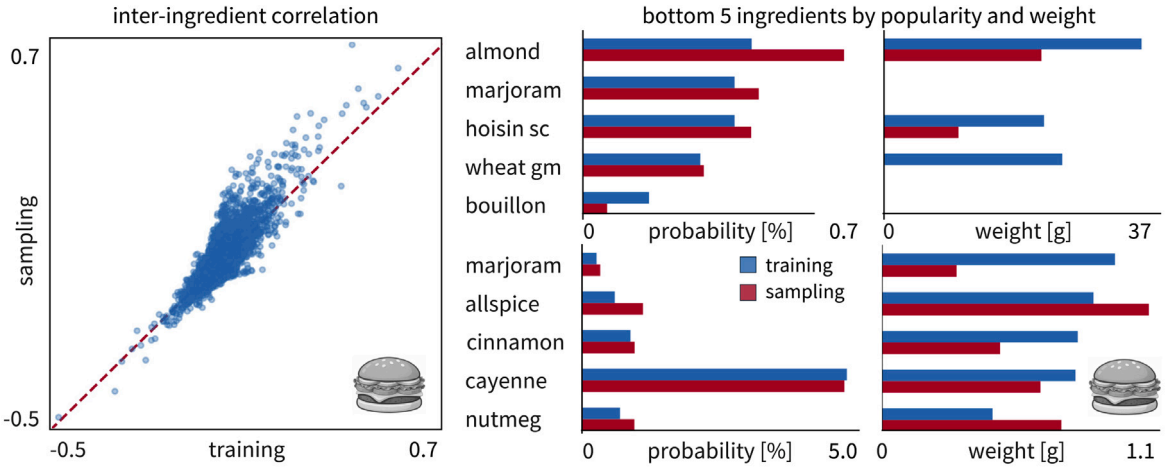

Fig. 14. Continuous diffusion in high-dimensional space. Comparison of 2260 training burgers and 1,000,000 samples generated by the learned continuous diffusion model. Inter-ingredient correlations between training and sampling across all 146 ingredients (left), occurrence probability of the bottom 5 least frequent ingredients (top right) and average ingredient weights for the least frequent ingredients by total weight (bottom right). The close agreement between training and sampling statistics, including pairwise correlations and rare ingredients, demonstrates that the model accurately captures both high-order dependencies and rare-event structure across the full distribution.

which uses an animal-meat–mushroom blend, scores comparably to all entirely animal-based burgers across key attributes ($p>0.05$), including overall liking (5.4 ± 1.6), flavor (5.4 ± 1.6), and texture (5.1 ± 1.6) which suggests that partial substitution of animal protein with mushroom does not substantially degrade perceived texture [54]. In a complementary texture profile analysis, a double compression test that mimics the process of chewing, the stiffnesses of the AI-generated burger patties range from 612 ± 171 kPa for the delicious burger 1 to 27 ± 18 kPa for the sustainable burger 1, and span the range of the classical Big Mac of 279 ± 130 kPa [55]. This agreement indicates that the generated configurations are not only statistically consistent, but also physically feasible in terms of measurable mechanical response. Taken together, these results demonstrate that diffusion-based generative models do not merely reproduce existing designs, but can discover entirely new, high-quality solutions that surpass established human-designed benchmarks. In the context of a design space of astronomical size and limited training data, this highlights the power of generative AI as a tool for high-dimensional discovery and data-driven innovation.

5. Discussion

The objective of this manuscript was to develop a unified generative framework that links discrete ingredient selection and continuous ingredient quantification through diffusion processes, and to connect diffusion-based generative AI to classical concepts in computational mechanics. We show that both formulations follow the same stochastic degradation and learned inversion principle,

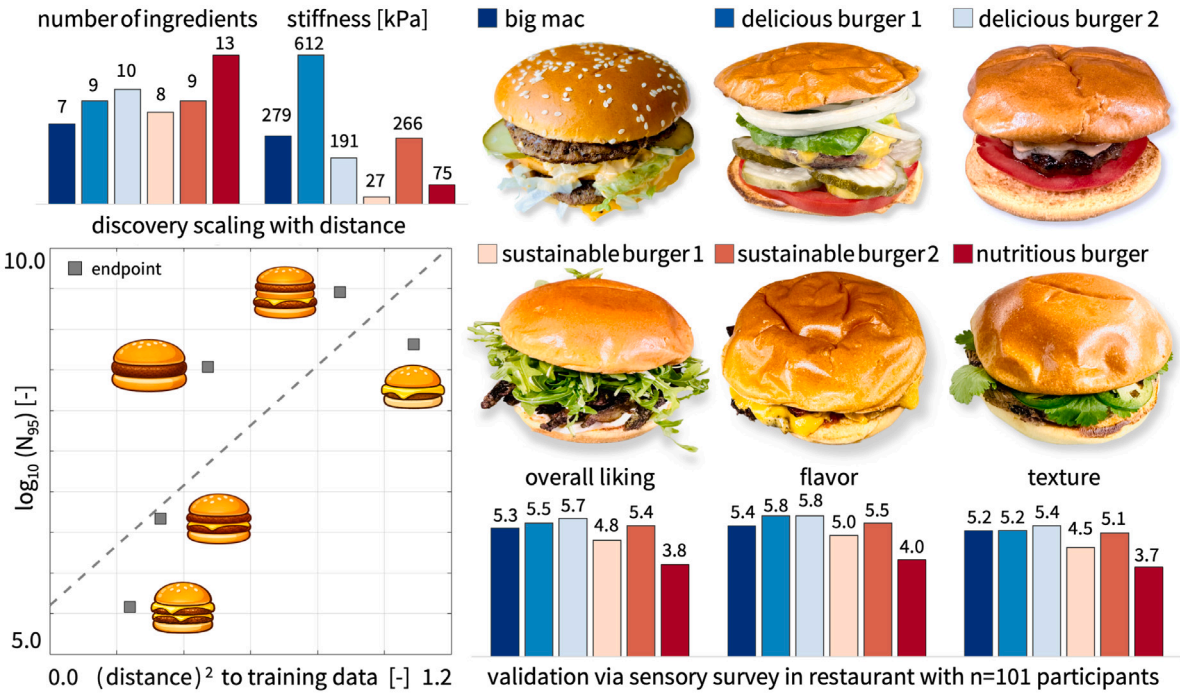

Fig. 15. Generating new burgers in high-dimensional space. Sampling complexity for discovering five target burgers, Double Cheeseburger, Mc Double, Quarter Pounder, Big Mac, and Cheeseburger, reported as number of sample trajectories required for 95% discovery vs. squared distance to training manifold (bottom left). Number of ingredients and patty stiffness of Big Mac and five AI-generated burgers (top left). Photographs of the prepared burgers, including the Big Mac, two delicious burgers, two sustainable burgers, and one nutritious burger (top right). Blinded sensory evaluation results from n = 101 restaurant participants rating overall liking, flavor, and texture on a 7-point Likert scale (bottom right). Three AI-generated burgers outperform the Big Mac in overall liking and flavor, and one AI-generated burger exceeds the Big Mac in texture. These results demonstrate that AI-driven generative design can produce novel food formulations that rival or exceed established commercial benchmarks in real-world sensory perception.

which enables the generation of novel burgers that preserve the structure of the training data while extending beyond the training manifold.

Analogy of discrete and continuous diffusion in low dimensions. Sections 2 and 3 highlight the direct analogy between *discrete* and *continuous* diffusion as two parallel formulations of the same generative principle in a *low-dimensional* setting. In the discrete case, burger configurations evolve on the finite state space $\{0,1\}^3$ via a Markov chain (Eq. (11)), admit a closed-form solution (Eq. (14)), and can be reversed exactly using Bayes' theorem (Eq. (18)), with sampling performed through categorical transitions. In the continuous case, ingredient weights evolve in $\mathbb{R}^3$ via an Ornstein–Uhlenbeck stochastic differential equation (Eq. (24)), admit a Gaussian closed-form solution (Eq. (25)), and are reversed through the score function (Eq. (30)), with trajectories generated via Euler–Maruyama discretization. In both settings, the three-ingredient benchmark makes the reverse dynamics analytically tractable and provides a unified and interpretable framework that connects discrete probabilistic transitions with continuous score-based diffusion. This unified view expands across discrete and continuous representations, forward and reverse processes, and low- and high-dimensional settings (Table 2).

Analogy of discrete and continuous diffusion in high dimensions. Sections 4.1 and 4.2 extend the analogy between *discrete* and *continuous* diffusion to *high-dimensional* generative modeling, where exact analytical solutions are no longer tractable and must be approximated through learning. In the discrete case, burger configurations evolve on the exponentially large state space $\{0,1\}^n$ via a factorized Markov chain (Eq. (35)), where forward transitions remain analytically defined (Eq. (37)), but reverse transitions become intractable due to the summation over all possible configurations (Eq. (38)). We therefore approximate the reverse process by a neural network that predicts the clean configuration (Eq. (40)), via sampling through learned categorical transitions (Eq. (41)). In the continuous case, ingredient weights evolve in $\mathbb{R}^n$ via a stochastic differential equation (Eq. (42)), where the forward process admits a closed-form Gaussian solution (Eq. (44)), but the reverse-time dynamics depend on the score function through the reverse stochastic differential equation (Eq. (43)), which is unknown in high dimensions and must be learned from data. We therefore approximate the reverse process by a neural network that predicts the underlying data structure through the score function (Eq. (46)), via sampling through integration of the learned stochastic dynamics. In both settings, high dimensionality transforms analytically tractable reverse dynamics into learning problems, where neural networks approximate either posterior distributions in discrete space (Eq. (40)) or score functions in continuous space (Eq. (46)). This establishes a unified generative framework in which discrete

Table 2
Unified analogy of diffusion models. Representations, directions, dimensionality, governing equations, and canonical references.

concept	discrete diffusion	continuous diffusion
example	ingredients	weights
state space	$\boldsymbol{x} \in \{0,1\}^n$ binary hypercube	$\boldsymbol{w} \in \mathbb{R}^n$, Euclidean space
forward diffusion	**Markov chain**	**Ornstein–Uhlenbeck process**
	Markov [56], Kolmogorov [57]	Ornstein–Uhlenbeck [43], Fokker–Planck [15,16]
	$p(\boldsymbol{x}_t\|\boldsymbol{x}_{t-1})$ (Eq. (35))	$\mathrm{d}\boldsymbol{w}_t = -\frac{1}{2}\beta_t \boldsymbol{w}_t \mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}\boldsymbol{B}_t$ (Eq. (42))
interpretation	random bit flips with flip rate β_t	Gaussian perturbation with diffusion rate β_t
entropy evolution	probability spreads over 2^n states	density spreads towards isotropic Gaussian
forward solution	closed form: $p(\boldsymbol{x}_t\|\boldsymbol{x}_0)$ (Eq. (14))	closed form Gaussian: $p(\boldsymbol{w}_t\|\boldsymbol{w}_0) = \mathcal{N}(\mu(t)\boldsymbol{w}_0, \sigma(t)\boldsymbol{I})$ (Eq. (44))
reverse diffusion	**Bayesian inversion**	**reverse-time Ornstein–Uhlenbeck process**
	Bayes [58]	Anderson [44]
	$p(\boldsymbol{x}_{t-1}\|\boldsymbol{x}_t)$ (Eq. (18))	$\mathrm{d}\boldsymbol{w}_t = [\frac{1}{2}\beta_t \boldsymbol{w}_t + \beta_t \nabla \log(p_t)]\mathrm{d}t + \sqrt{\beta_t}\mathrm{d}\tilde{\boldsymbol{B}}_t$ (Eq. (43))
interpretation	probability update on graph nodes	drift along score field $\nabla \log(p_t)$
low-dim reverse solution	exact transitions for finite states	exact score of Gaussian mixture
high-dim reverse solution	intractable sum over 2^{146} states	intractable sum over training data
learning	**denoising diffusion probabilistic model**	**score-based generative model**
	Ho et al. [12]	Song et al. [13]
	$p_\theta(\boldsymbol{x}_0\|\boldsymbol{x}_t)$ (Eq. (40))	$s_\theta(\boldsymbol{w}, t) \approx \nabla_{\boldsymbol{w}} \log(p_t(\boldsymbol{w}))$ (Eq. (46))
interpretation	learn posterior/logits	learn score function
sampling	sequential categorical updates: $\boldsymbol{x}_{t-1} \sim p_\theta(\boldsymbol{x}_{t-1}\|\boldsymbol{x}_t)$	Euler–Maruyama integration: [45] $\boldsymbol{w}_{t-\Delta t} = \boldsymbol{w}_t + [\frac{1}{2}\beta_t \boldsymbol{w}_t + \beta_t\, s_\theta(\boldsymbol{w}_t, t)]\Delta t + \sqrt{\beta_t \Delta t}\,\boldsymbol{\xi}$
unifying view	**learning to reverse entropy-increasing stochastic processes**	
generative role	select ingredients $\boldsymbol{x}$	assign weights $\boldsymbol{w}$
geometric interpretation	graph transport on hypercube	continuum transport on energy landscape
mechanics analogy	probability flux on discrete graph	probability density flow in continuous field
unifying principle	learn reverse Markov dynamics	learn reverse stochastic dynamics

and continuous diffusion differ only in representation, but share the same underlying principle of learning to reverse stochastic degradation. This transition becomes explicit when contrasting the analytically tractable low-dimensional case with the learned high-dimensional case for both discrete and continuous diffusion (Table 2).

From burgers to matter. Our results position *generative AI for burgers* as a direct analogue of *generative AI for matter*, as recently demonstrated by MatterGen, a generative framework that designs inorganic materials with targeted properties [9]. In our framework, ingredients play the role of chemical elements, and the ingredient table mirrors the periodic table as a discrete basis for constructing complex systems. Both problems share the same combinatorial explosion: selecting subsets and assigning weights defines an astronomically large design space that vastly exceeds available training data. Diffusion models address this challenge by learning to generate structured configurations – recipes or crystal structures – that satisfy underlying statistical and physical constraints while remaining novel [59]. This establishes a common paradigm where generative models enable inverse design: burgers can be optimized for taste, nutrition, and sustainability, just as materials can be designed for stability, functionality, and resource constraints.

From food science to computational mechanics. We can now witness the first applications of diffusion models in the field of computational mechanics: solving inverse problems and uncertainty quantification in physics-based systems [60], generating of constitutive laws under physics constraints [52], forecasting and reconstructing the dynamics of incompressible flows [61], reconstructing complex microstructures in materials engineering [62], generating shell structures with targeted stress–strain responses [63], and generating energy-absorbing metamaterials and structures [64]. In contrast to these application-specific formulations, our framework provides a unified abstraction that links *discrete* and *continuous* design spaces through a common diffusion-based transport mechanism. Rather than targeting a single task—such as reconstruction, inference, or constitutive generation, we interpret diffusion models as a general operator for transforming simple priors into structured matter and connect generative AI directly to the principles of computational mechanics. While the present study does not explicitly impose physical constraints during training, we can incorporate physical feasibility by conditioning the generative process on governing laws [59] or by constructing representations that satisfy the physics a priori [52]. This parallel highlights a unifying paradigm: generative AI acts as a data-driven engine for high-dimensional discovery that transforms discrete building blocks – ingredients or elements – into optimized, functional matter across domains. These developments build on a common theoretical foundation in denoising diffusion probabilistic models [12], score-based generative modeling [13], and automated model discovery in mechanics [65], and extend from images [66] to proteins [7] and inorganic materials [67], which underscores the generality of diffusion-based inverse design across domains. In both discrete and continuous settings, training minimizes a Kullback–Leibler divergence [68] between forward and reverse processes, which quantifies a mismatch in probability flux, and admits an interpretation as a free-energy functional in the sense of variational diffusion [25].

6. Conclusion

We have introduced a unified diffusion-based framework for generative design that combines discrete ingredient selection with continuous ingredient quantification. The formulation establishes a direct correspondence between Markov chains and stochastic differential equations, and between Bayesian inversion and score-based learning. Both models generate structured samples that recover the training data and produce novel configurations beyond the training manifold. Discovery probability decays rapidly with geometric distance in ingredient space, which identifies distance as a governing variable for exploration in high dimensions. These results establish diffusion models as a principled tool for design in spaces where combinatorial complexity prohibits exhaustive search. Beyond its educational value, this framework provides a transparent testbed for analyzing exploration, discovery, and inverse design in high-dimensional spaces. It offers a mechanics-based language to quantify how generative models navigate combinatorial design spaces and to develop new strategies for property-targeted discovery. At its core, our framework builds on concepts that are fundamental to mechanics – stochastic processes, diffusion, and inverse problems – and recasts them as engines for generative modeling. While we use burgers as a minimal and interpretable benchmark for structured matter, generally, diffusion-based methods in machine learning extend directly to materials, biological systems, and other high-dimensional design spaces. This work reframes diffusion not only as a model of physical processes, but as a general mechanism for the design of matter and materials.

CRediT authorship contribution statement

Vahidullah Tac: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ellen Kuhl:** Writing – review & editing, Writing – original draft, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank Executive Chef Justin Schneider for his culinary expertise in creating preparation instructions, Caroline Cotto from NECTAR at Food System Innovations for stimulating discussions, and Alice Wistar and Alex Weissman from Palate Insights for performing the costumer survey. We acknowledge access to the Stanford Marlowe Computing Platform for high performance computing. This research was supported by the Schmidt Science Fellowship in partnership with the Rhodes Trust to Vahidullah Tac, and by the Stanford Bio-X Snack Grant 2025, the Stanford SDSS Accelerator Grant 2025, the NSF CMMI Award 2320933, and the ERC Advanced Grant 101141626 to Ellen Kuhl.

Data availability

Our source code, data, and examples are available at https://github.com/LivingMatterLab/AI4Food.

References

[1] A.F. Smith, Hamburger: A Global History, Reaktion Books, London, 2008.
[2] B.M. Bohrer, An investigation of the formulation and nutritional composition of modern meat analogue products, Food Sci. Hum. Wellness 8 (2019) 320–329.
[3] A.D. Unruh, J.J. Kastner, J.R. Jenott, S.E. Gragg, Handling of hamburgers and cooking practices, in: Food Hygiene and Toxiology in Ready-to-Eat Foods, 2016, pp. 107–122, (Chapter 7).
[4] A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K.A. Persson, Commentary: The materials project: A materials genome approach to accelerating materials innovation, APL Mater. 1 (2013) 011002.
[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, NeurIPS, in: Advances in Neural Information Processing Systems, vol. 27, 2014, pp. 2672–2680.
[6] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, 2014, http://dx.doi.org/10.48550/arXiv.1312.6114, arXiv.
[7] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Zidek, A. Potapenko, A. Bridgland, C. Meyer, S.A.A. Kohl, A.J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A.W. Senior, K. Kavukcuoglu, P. Kohli, D. Hassabis, Highly accurate protein structure prediction with AlphaFold, Nature 596 (2021) 583–589.
[8] K.T. Butler, D.W. Davies, H. Cartwright, O. Isayev, A. Walsh, Machine learning for molecular and materials science, Nature 559 (2018) 547–555.
[9] C. Zeni, R. Pinsler, D. Zügner, A. Fowler, M. Horton, X. Fu, Z. Wang, A. Shysheya, J. Crabbe, S. Ueda, R. Sordillo, L. Sun, J. Smith, B. Nguyen, H. Schulz, S. Lewis, C.W. Huang, Z. Lu, Y. Zhou, H. Yang, H. Hao, J. Li, C. Yang, W. Li, R. Tomioka, Tian Xie T., A generative model for inorganic materials design, Nature 639 (2025) 624–632.
[10] R. Gomez-Bombarelli, J.N. Wei, D. Duvenaud, J.M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T.D. Hirzel, R.P. Adams, A. Aspuru-Guzik, Automatic chemical design using a data-driven continuous representation of molecules, ACS Cent. Sci. 4 (2018) 268–276.
[11] J. Sohl-Dickstein, E.A. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in: Proceedings of the 32nd International Conference on Machine Learning, vol. 37, PMLR, 2015, pp. 2256–2265.

[12] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, NeurIPS, Advances in Neural Information Processing Systems, vol. 33 (2020) 6840–6851, http://dx.doi.org/10.48550/arXiv.2006.11239, arXiv.
[13] Y. Song, J. Sohl-Dickstein, D.P. Kingma, A. Kumar, S. Ermon, B. Poole, Score-based generative modeling through stochastic differential equations, in: International Conference on Learning Representations, 2021, http://dx.doi.org/10.48550/arXiv.2011.13456, arXiv.
[14] H. Risken, The Fokker–Planck Equation: Methods of Solution and Applications, Springer, Berlin, 1996.
[15] A.D. Fokker, Die mittlere Energie rotierender elektrischer Dipole im Strahlungsfeld, Ann. Phys. 348 (1914) 810–820.
[16] M. Planck, Über einen Satz der statistischen Dynamik und seine Erweiterung in der Quantentheorie, Sitzungsber. Preuss. Akad. Wiss. Zu Berl. 24 (1917) 324–341.
[17] C.W. Gardiner, Handbook of Stochastic Methods for Physics, Chemistry, and the Natural Sciences, Springer, Berlin, 2009.
[18] A Fick, Ueber diffusion, Ann. Phys., Lpz. 94 (1) (1855) 59–86.
[19] J. Fourier, Théorie Analytique de la Chaleur, Firmin Didot, Paris, 1822.
[20] G. Kirchhoff, Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird, Ann. Phys. 72 (1847) 497–508.
[21] F.R.K. Chung, Spectral graph theory, in: CBMS Regional Conference Series in Mathematics, no. 92, American Mathematical Society, 1997.
[22] J.W. Cahn, J.E. Hilliard, Free energy of a nonuniform system. I. Interfacial free energy, J. Chem. Phys. 28 (1958) 258–267.
[23] L. Onsager, Reciprocal relations in irreversible processes, Phys. Rev. 37 (1931) 405–426.
[24] L. Ambrosio, N. Gigli, G. Savaré, Gradient Flows in Metric Spaces and in the Space of Probability Measures, Birkhäuser, Basel, 2008.
[25] R. Jordan, D. Kinderlehrer, F. Otto, The variational formulation of the Fokker–Planck equation, SIAM J. Math. Anal. 29 (1998) 1–17.
[26] V. Taç, C. Gardner, E. Kuhl, Generative artificial intelligence creates delicious, sustainable, and nutritious burgers, 2026, http://dx.doi.org/10.48550/arXiv.2602.03092, arXiv.
[27] C. Villani, Optimal Transport: Old and New, Springer, Berlin, 2009.
[28] B. Datta, M.J. Buehler, Y. Chow, K. Gligoric, D. Jurafsky, D.L. Kaplan, R. Lesema-Amaro, G. Del Missier, L. Neidhardt, K. Pichara, B. Sanchez-Lengeling, M. Schlangen, S.R. St Pierre, I. Tagkopoulos, A. Thomas, N. Watson, E. Kuhl, AI for sustainable food futures: from data to dinner, 2025, http://dx.doi.org/10.48550/arXiv.2509.21556, arXiv.
[29] L.O. Figura, A.A. Teixeira, Food Physics: Physical Properties - Measurements and Applications, Springer Nature, Switzerland, 2023.
[30] F. Li, J. Youn, T. Chan, P. Gupta, A. Yoo, M. Gunning, K. Ni, I. Tagkopoulos, A unified knowledge graph linking foodomics to chemical-disease networks and flavor profiles, npj Sci. Food 10 (2026) 33.
[31] S.R. St. Pierre, E.C. Darwin, D. Adil, M.C. Aviles, A. Date, R.A. Dunne, Y. Lall, M. Parra Vallecillo, V.A. Perez Medina, K. Linka, M.E. Levenston, E. Kuhl, The mechanical and sensory signature of plant-based and animal meat, npj Sci. Food 8 (2024) 94.
[32] M. Al-Sarayreh, M. Gomes Reis, A. Carr, M. Martinis dos Reis, Inverse design and AI/Deep generative networks in food design: A comprehensive review, Trends Food Sci. Tech. 138 (2023) 215–228.
[33] A. Datta, B. Nicolai, O. Vitrac, P. Verboven, F. Erdogdu, F. Marra, F. Sarghini, C. Koh, Computer-aided food engineering, Nat. Food 3 (2022) 894–904.
[34] M. Gunning, M.P. Serantes Laforgue, J.X. Guinard, I. Tagkopoulos, AI-driven prediction of consumer liking of coffee from sensory data, npj Sci. Food (2026) http://dx.doi.org/10.1038/s41538-026-00779-7, in press.
[35] I. Tagkopoulos, S.F. Brown, X. Liu, Q. Zhao, T.I. Zohdi, J.M. Earles, N. Nitin, D.E. Runcie, D.G. Lemay, A.D. Smith, P.C. Ronald, H. Feng, D.G. Youtsey, Special report: AI institute for next generation food systems (AIFS), Comp. Elect. Agri. 196 (2022) 106819.
[36] S.D. van den Bedem, E. Kuhl, C. Cotto, Open-source benchmarking of plant-based and animal meats, 2026, http://dx.doi.org/10.48550/arXiv/2603.03370, arXiv.
[37] T. Zohdi, A voxel-based machine-learning digital-oven-twin for precise cooking, Comput. Mech. 75 (2024) 1501–1518.
[38] E. Kuhl, AI for food: Accelerating and democratizing discovery and innovation, npj Sci. Food 9 (2025) 82.
[39] NotCo, Latent space method of generating food formulas, 2021, US 10, 915, 818.
[40] NotCo, Neural network method of generating food formulas, 2021, US 10, 957, 424.
[41] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (1948) 379–423, 623–656.
[42] A. Schafer, E.C. Mormino, E. Kuhl, Network diffusion modeling explains longitudinal tau PET data, Front. Neurosci. 14 (2020) 566876.
[43] G.E. Uhlenbeck, L.S. Ornstein, On the theory of the Brownian motion, Phys. Rev. 36 (1930) 823–841.
[44] B.D.O. Anderson, Reverse-time diffusion equation models, Stochastic Process. Appl. 12 (1982) 313–326.
[45] G. Maruyama, On the transition probability functions of the Markov process, in: Nat. Sci. Report, vol. 5, Ochanomizu University, 1954, pp. 10–20.
[46] H.A. Kramers, Brownian motion in a field of force and the diffusion model of chemical reactions, Physica 7 (1940) 284–304.
[47] L. Onsager, S. Machlup, Fluctuations and irreversible processes, Phys. Rev. 91 (1931) 1505–1512.
[48] S. Arrhenius, Über die reaktionsgeschwindigkeit bei der inversion von rohrzucker durch säuren, Z. Phys. Chem. 4 (1889) 226–248.
[49] P. Langevin, Sur la théorie du mouvement brownien, C. R. Acad. Sci. (Paris) 146 (1908) 530–533.
[50] M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, J. Wortman Vaughan, Argmax flows and multinomial diffusion: learning categorical distributions, NeurIPS, in: Advances in Neural Information Processing Systems, vol. 34, 2021, pp. 12454–12465.
[51] J. Pidstrigach, Score-based generative models detect manifolds, NeurIPS, in: Advances in Neural Information Processing Systems, vol. 35, 2022, pp. 35852–35865.
[52] V. Taç, M. Rausch, I. Bilionis, F. Sahli Costabal, A. Buganza Tepole, Generative hyperelasticity with physics-informed probabilistic diffusion fields, Eng. Comp. 41 (2024) 51–69.
[53] T. Karras, M. Aittala, T. Aila, S. Laine, Elucidating the design space of diffusion-based generative models, NeurIPS, in: Advances in Neural Information Processing Systems, vol. 35, 2022, pp. 26565–26577.
[54] S.R. St. Pierre, A.O. Koosis, N. Zhang, E. Kuhl, The meatball matchup: Plant vs animal proteins on campus, Food Res. Int. 240 (2026) 119631.
[55] V. Taç, A.O. Koosis, E. Kuhl, Texture independently drives liking in AI-generated alternative protein burgers, Foods 15 (2026) 2026.
[56] A.A. Markov, Extension of the law of large numbers to dependent events, Bull. Soc. Phys. Math. Kazan Russ. 2 (1906) 155–156.
[57] A.N. Kolmogorov, Über die analytischen Methoden in der Wahrscheinlichkeitsrechnung, Math. Ann. 104 (1931) 415–458.
[58] T. Bayes, An essay towards solving a problem in the doctrine of chances, Philos. Trans. R. Soc. Lond. 53 (1763) 370–418.
[59] J.H. Bastek, W.C. Sun, D.M. Kochmann, Physics-informed diffusion models, 2025, http://dx.doi.org/10.48550/arXiv.2403.14404, arXiv.
[60] A. Dasgupta, A.M. da Cunha, A. Fardisi, M. Aminy, B. Binder, B. Shaddy, A.A. Oberai, Unifying and extending diffusion models through PDEs for solving inverse problems, Comput. Methods Appl. Mech. Engrg. 448 (2026) 118431.
[61] H. Gao, S. Kaltenbach, P. Koumoutsakos, Generative learning of the solution of parametric partial differential equations using guided diffusion models and virtual observations, Comput. Methods Appl. Mech. Engrg. 435 (2025) 117654.
[62] C. Düreth, P. Seibert, D. Rücker, S. Handford, M. Kästner, M. Gude, Conditional diffusion-based microstructure reconstruction, Mat. Today Comm. 35 (2023) 105608.
[63] L. Zheng, S. Kumar, D.M. Kochmann, Algebraic language models for inverse design of metamaterials via diffusion transformers, Nat. Mach. Intel. 8 (2026) 628–640.
[64] H. Wang, Z. Du, F. Feng, Z. Kang, S. Tang, X. Guo, DiffMat: Data-driven inverse design of energy-absorbing metamterials using diffusion model, Comput. Methods Appl. Mech. Engrg. 432 (2024) 117440.

[65] K. Linka, E. Kuhl, A new family of constitutive artificial neural networks towards automated model discovery, Comput. Methods Appl. Mech. Engrg. 403 (2023) 115731.
[66] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recogn., 2022, pp. 10684–10695, http://dx.doi.org/10.48550/arXiv.2112.10752, arXiv.
[67] J.L. Watson, D. Juergens, N.R. Bennett, B.L. Trippe, J. Yim, H.E. Eisenach, W. Ahern, A.J. Borst, R.J. Ragotte, L.F. Milles, B.I.M. Wicky, N. Hanikel, S.J. Pellock, A. Courbet, W. Sheffler, J. Wang, P. Venkatesh, I. Sappington, S.V. Torres, A. Lauko, V. De Bortoli, E. Mathieu, S. Ovchinnikov, R. Barzilay, T.S. Jaakkola, F. DiMaio, M. Baek, D. Baker, De novo design of protein structure and function with RFdiffusion, Nat. 620 (2023) 1089–1099.
[68] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1951) 79–86.