DISCRETE FLOW MATCHING FOR REGULATORY DNA SEQUENCE DESIGN

Muhammad Hashaam & Maria Poptsova
Institute of Artificial Intelligence and Digital Sciences
Faculty of Computer Science
HSE University
Moscow, Russia
mpoptsova@hse.ru.edu

ABSTRACT

Flow matching and diffusion models have achieved strong performance in continuous data domains, but extending these methods to discrete biological sequences remains challenging. Recently, discrete generative frameworks have been proposed to address this limitation. Discrete Flow Matching (DFM) is a generative paradigm designed specifically for modeling discrete state spaces without continuous relaxation. While DFM has shown promising results in domains such as protein design and text generation, its applicability to regulatory DNA sequence design remains underexplored. In this work, we investigate the use of Discrete Flow Matching for DNA sequence generation, focusing on promoter and enhancer design tasks. We benchmark our approach on three genomic datasets, including human promoters and enhancers from human melanoma and Drosophila brain tissues. We evaluate generation quality using both distributional metrics, such as Fréchet Biological Distance, and functional metrics based on predictive regulatory models. Our results show that DFM achieves competitive or superior performance compared to existing diffusion- and flow-based methods, particularly in unconditional enhancer generation and conditional promoter design. By operating directly on discrete nucleotide sequences, DFM avoids relaxation-induced artifacts and preserves biological sequence structure. These findings demonstrate that Discrete Flow Matching is a principled and effective framework for regulatory DNA sequence design. At the same time, our results reveal clear limitations of likelihood-based discrete flows in sampling rare, high-activity sequences, highlighting challenges in modeling extreme regions of complex activity landscapes To our knowledge, this work provides the first systematic evaluation of Discrete Flow Matching for regulatory DNA sequence design across both promoter and enhancer settings.

1 BACKGROUND

DNA sequence design is a central problem in computational biology, with applications in gene regulation, synthetic biology, and therapeutic development. Designing sequences with desired functional properties is challenging due to the complex, high-dimensional, and poorly understood structure of regulatory DNA, making data-driven generative modeling approaches particularly appealing Sarkar et al. (2024).

Early work on DNA sequence generation primarily relied on variational autoencoders (VAEs) and generative adversarial networks (GANs), which demonstrated the feasibility of learning generative models over regulatory sequences such as promoters and enhancers Killoran et al. (2017); Gupta & Zou (2019). These approaches enabled both unconditional generation and property-guided optimization, but often suffered from training instability, mode collapse, or limited scalability.

More recently, diffusion-based generative models have emerged as a powerful alternative for biological sequence modeling Ho et al. (2020). By progressively transforming noise into data through a sequence of denoising steps, diffusion models offer improved training stability and support flexible

conditional generation strategies. Extensions of diffusion models to discrete domains have enabled their application to DNA sequence generation, including regulatory element modeling Avdeyev et al. (2023).

Flow matching models provide a closely related but distinct framework, learning a deterministic continuous-time transport from a base distribution to the data distribution via an ordinary differential equation Lipman et al. (2022); Chen et al. (2023). Recent work has applied flow-based methods to genomic sequence design, including Dirichlet and Fisher flow matching formulations Stark et al. (2024); Davis et al. (2024). However, these approaches typically rely on continuous relaxations of discrete sequences.

In this work, we study discrete flow matching (DFM) Gat et al. (2024) as a fully discrete generative modeling paradigm for regulatory DNA. We evaluate its performance on promoter and enhancer datasets, focusing on sample quality and functional fidelity as measured by downstream predictive models. Our contributions are:

- We apply discrete flow matching to the generation of regulatory DNA sequences, including promoters and enhancers.
- We extend DFM to conditional sequence generation using classifier-free guidance.
- We analyze limitations of DFM for conditional generation and identify directions for future research.

2 METHODS

2.1 DATASETS

We evaluated DFM on three DNA sequence datasets: two enhancer datasets and one promoter dataset. Specifically, we used two enhancer datasets that contain DNA sequences from human melanoma cells Atak et al. (2021) and fruit fly brain cells Janssens et al. (2022).

Each enhancer sequence is associated with a cell type label that indicates the specific cell type in which the enhancer is active. Each sequence is 500 base pairs long. Melanoma data consist of 70892 train sequences, 9012 test sequences, and 8966 validation sequences. Fly brain data consist of 83726 train sequences, 10434 test sequences, and 10505 validation sequences.

The promoter data set Hon et al. (2017) consists of 100,000 human promoter sequences and the corresponding transcription initiation signal profiles. Transcription initiation signal profiles reflect transcription initiation activity at every sequence position and are obtained from CAGE experiments Consortium et al. (2014). Each sequence is 1024 base pair long and centered at the annotated transcription start site position. The dataset is split into 88570 train sequences, 7497 test sequences, and 3933 validation sequences.

2.2 NEURAL ARCHITECTURE

All models in this work use the same architectural structure: a 20-layer 1-D dilated convolutional network with kernel size 9 and dilation pattern 1,1,4,16,64 repeated four times. At each layer, we add a Gaussian Fourier time embedding projected through a dense layer. Each convolutional block uses normalization, nonlinear activation, dropout, and a residual connection when dimensions match. The network ends with two 1×1 convolutions with GELU activation that map features to nucleotide logits. The enhancer models use a channel width of 128 and contain approximately 3.7M trainable parameters, while the promoter models use 256 channels with approximately 13.3M trainable parameters. This choice reflects the shorter sequence length and reduced complexity of enhancer datasets compared to promoter sequences.

We use this setup for fair comparison with previous models, as it was used by all the previous approaches for regulatory DNA sequence design.

2.3 Evaluation Metrics

To evaluate the quality of the generated sequences, we use established metrics including Perplexity, the Fréchet Biological Distance (FBD) and Mean Squared Error (MSE).

MSE is used to evaluate the quality of the generated promoter sequences. For this evaluation we need another deep learning model called SEI Chen et al. (2022). SEI is a predictive model that can predict active promoter from a DNA sequence (based on chromatin mark H3K4me3 predictions) Chen et al. (2022) . First, SEI is used to predict the transcription activity of original human sequence. Conditioned on the original transcription profile, a sequence is generated using the diffusion model. The SEI model is then used to predict the transcription activity of the generated sequence. MSE is then used to compare the predicted activity of the generated sequence to that of the original sequence. If the generated sequence is of good quality and adheres to the conditional signal, the MSE will be low.

The metric used to evaluate the enhancer sequences is the Fréchet Biological Distance (FBD). FBD similar to Fréchet Inception Distance Heusel et al. (2017), measures the distance between the distribution of the real and generated sequences. The smaller the FBD, the closer the distributions, which means that the generated data is similar to the original data. To find the FBD, we use a separate trained classifier that predicts the cell type label and use its hidden representations as vector embeddings. Using this classifier, we get a 128 dimensional vector embedding for each sequence. We calculate the mean vector and the covariance matrix of the embeddings of the ground truth sequences and generated sequences. The FBD is then calculated using the equation.

$$FBD = \|\mu_{\text{real}} - \mu_{\text{gen}}\|^2 + \text{Tr}\Big(\Sigma_{\text{real}} + \Sigma_{\text{gen}} - 2\left(\Sigma_{\text{real}}\Sigma_{\text{gen}}\right)^{1/2}\Big)$$

2.4 Baseline Models

2.4.1 Dirichlet Diffusion Score Model (Promoters)

The Dirichlet Diffusion Score Model Avdeyev et al. (2023) is one of the earliest diffusion-based approaches for DNA sequence generation. Their dataset consists of human promoter sequences and the corresponding transcription initiation signal profiles. Using this dataset, they train a diffusion model that generates promoter sequences conditioned on the transcription initiation signal profile.

2.4.2 Dirichlet Flow Matching (enhancers)

Dirichlet Flow Matching Stark et al. (2024) extends the Dirichlet Diffusion Score Model to the flow matching framework. In addition to promoter sequence generation, they also use their model to generate enhancer DNA sequences. Their dataset consists of 500 bp long enhancer sequences paired with cell-type labels. Using this dataset, they build unconditional and conditional generative models for enhancers. The unconditional model generates enhancers without conditioning on cell-type labels. The conditional model uses the cell-type label as an additional input and generates cell-type-specific enhancer sequences.

2.5 Our Model: Discrete Flow Matching

Discrete Flow Matching (DFM) Gat et al. (2024) is a recent generative modeling framework designed for discrete data. Previous flow matching models, such as Dirichlet Flow Matching, embed discrete sequences into a continuous space and relax them back into a discrete representation. This relaxation can lead to information loss. Discrete flow matching develops a flow matching algorithm for time-continuous Markov processes on discrete state spaces, also known as Continuous Time Markov Chains (CTMC). This advancement allows the possibility of direct modeling of discrete data such as DNA sequence, using flow matching.

Discrete Flow Matching learns a velocity field directly over the discrete state space, without embedding sequences into a continuous domain. This direct modeling approach, originally developed for text, is well-suited for biological data such as DNA sequences.

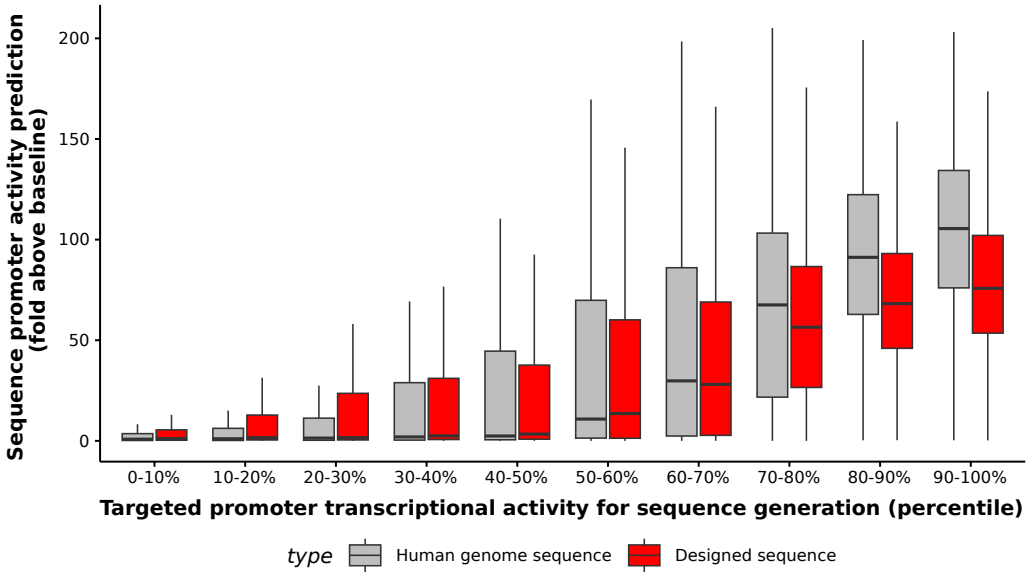

Figure 1: Comparison of predicted promoter activities for human genome sequences versus designed sequences. Generated sequences are grouped by the targeted promoter activity level (x-axis). Y-axis shows predicted promoter activity (average H3K4me3 prediction across cell types), divided by baseline prediction (0.004138) for average genomic sequences.

We use polynomial convex scheduler with mixture discrete probability path. The loss function is Generalized KL loss. We generate new samples by solving ODE using mixture discrete Euler solver.

3 Results

3.1 Promoter sequence generation

We trained our Discrete Flow Matching model for conditional promoter sequence generation for 500 epochs. The dataset consists of 1024 bp human promoter sequences paired with their transcription initiation signal profiles. Each sequence is one-hot encoded and concatenated with its corresponding transcription signal. The source distribution is a masked version of the input sequence, and the model learns to progressively unmask tokens over time.

For evaluation, we use the held-out test set. Generation begins by concatenating the masked initial sequence with the transcription signal, and the model gradually unmasks tokens along the flow trajectory. We evaluate generated promoter sequences using the SEI model, which predicts the transcription activity of a given promoter.

Figure 1 shows that for different levels of promoter activity, the generated sequence has predicted promoter activity comparable to that of the human genome sequence. Figure 2 shows the similarity of the promoter activity distribution between the original and generated sequences. Figure 3 shows GC content distribution of real and generated sequences. Table 1 shows the mean squared error (MSE) between the predicted transcription initiation signal from the generated sequence and that of the original sequence. Our model achieves MSE that is equal Fisher Flow Matching.

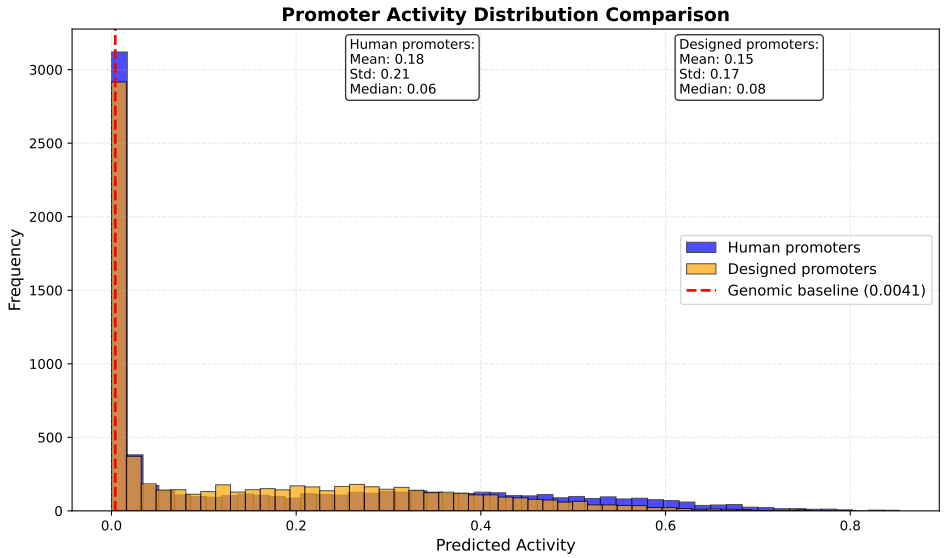

Figure 2: Distribution of predicted promoter activities for human and designed sequences. Both distributions show similar ranges, with the majority of predictions above baseline (red dashed line), indicating functional promoter activity.

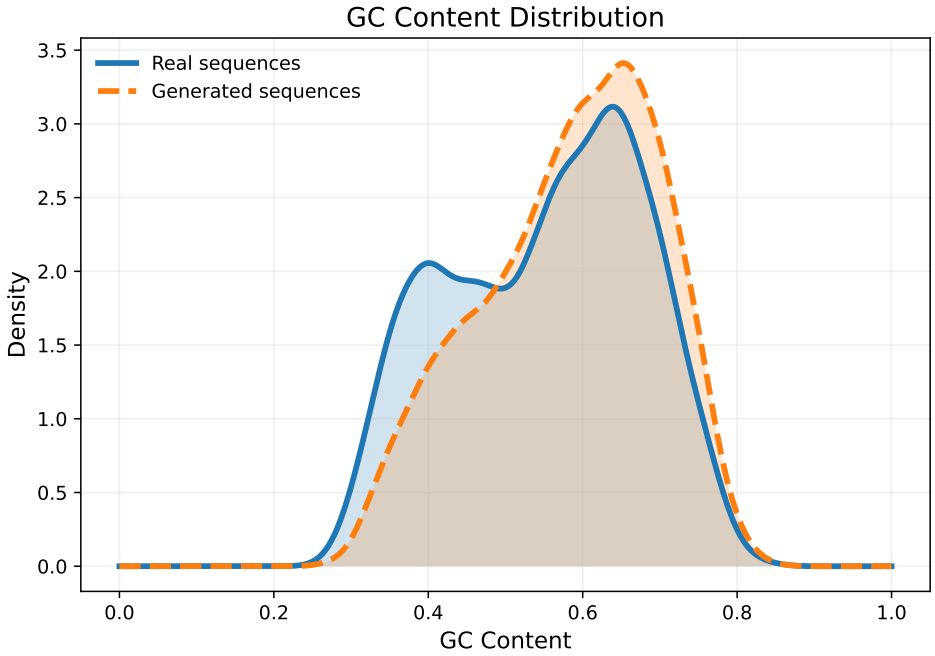

Figure 3: GC content distribution of 7497 human and designed sequences.

3.2 Enhancer sequence generation

We trained the unconditional and conditional generation model on enhancer sequences from Fly brain and human melanoma data with the discrete flow matching approach. We tested the quality of the model using the FBD metric.

Table 1: Mean squared error of the predicted transcription signal of the generated promoter DNA sequences over the test data. NFE refers to number of function evaluations

Model	MSE	NFE
Bit Diffusion	.040	100
DDSM	.033	100
Language Model	.034 ± 0.001	1024
Dirichlet FM	.034 ± 0.004	100
Fisher FM	**.029 ± 0.001**	100
Discrete FM (ours)	**.029 ± 0.001**	100

For Discrete FM, we calculate mean and standard deviation for 5 generations with different seeds. Metrics for other models are taken from Davis et al. (2024)

3.3 Unconditional generation

We train the unconditional enhancer sequence generation model for 1450 epochs. We show the FBD results under identical evaluation conditions in Table 2. The results show that our unconditional enhancer generation model achieves lower FBD than prior flow-based models on the melanoma dataset. For Fly brain dataset our model achieves better performance than Dirichlet Flow matching but little worse compared to Fisher Flow Matching on test data, while on train data Discrete FM achieves lowest FBD indicating strong capacity to model the enhancer distribution.

Table 2: FBD between the generated sequences and real enhancer sequences on the test split. NFE refers to number of function evaluations

Model	Melanoma FBD	Fly Brain FBD	NFE
Random Sequence	619.0 ± 0.8	832.4 ± 0.3	100
Language Model	35.4 ± 0.5	25.7 ± 1.0	500
Dirichlet FM	7.3 ± 1.2	6.8 ± 1.8	100
Fisher FM	27.5 ± 2.6	**3.8 ± 0.3**	100
Discrete FM	**3.9 ± 0.13**	4.5 ± 0.3	100

For Discrete FM, we calculate mean and standard deviation for 5 generations with different seeds. Metrics for other models are taken from Davis et al. (2024)

3.4 Conditional generation

For conditional generation, we train our model on Fly brain enhancer data conditioned on the cell type label. Each enhancer sequence is one-hot encoded, and conditioning is performed using a learned embedding of the cell-type label. This embedding is injected into every layer in parallel with the Gaussian Fourier time embedding.

During training we apply classifier-free guidance by randomly replacing the conditioning label with a null token with probability of 0.3. At inference time, we generate sequences by starting from a masked source distribution and iteratively unmasking tokens under the Discrete Flow Matching dynamics. We generate conditional and unconditional logits and extrapolate them with classifier free guidance scale of 3.

We generate sequences conditioned on 3 different cell type labels and we compare the quality of our model with Dirichlet flow matching model. We generate data with 2 kinds of initial distributions, uniform and masked initial distribution. The results indicate that DFM supports conditional generation, though performance remains slightly below Dirichlet Flow Matching.

4 Discussion

We demonstrated that Discrete Flow Matching (DFM) can be successfully applied to DNA sequence design across both enhancer and promoter settings. Across datasets, DFM achieves strong distribu-

Table 3: FBD for conditional enhancer generation task for fly brain dataset. Class 1, class 2, class 3 refer to different cell type labels.

Model	class 1	class 2	class 3	NFE
Dirichlet FM	**56.88**	83.56	**86.9**	100
Discrete FM (Masked)	86	**62.5**	155.5	100
Discrete FM (uniform)	66.7	80.6	98.8	100

We run Dirichlet FM and Discrete FM model conditioned on the specified cell types and measure cell type specific FBD on train split.

tional fidelity as measured by FBD and MSE, indicating its ability to capture the global structure of regulatory sequence space. In unconditional enhancer generation, DFM matches or improves upon prior flow-based approaches, while in conditional promoter generation it remains competitive with state-of-the-art likelihood-based models. These results establish discrete flow matching as a viable and principled framework for biological sequence modeling.

A key reason for the strong performance of DFM in DNA sequence modeling lies in its compatibility with the discrete nature of genomic data. Unlike continuous relaxations, discrete flow matching operates directly on nucleotide tokens, avoiding relaxation error introduced by continuous approximations. This allows the model to preserve base composition and local sequence statistics more faithfully, without the implicit Dirichlet smoothing present in prior approaches. Furthermore, the parallel generation inherent to flow-based models aligns well with regulatory grammar, where multiple motifs and dependencies act simultaneously rather than sequentially. Together, these properties enable DFM to model both local and long-range dependencies in regulatory DNA while maintaining biologically realistic sequence statistics.

Despite these strengths, our analysis reveals important limitations. While DFM achieves strong performance in unconditional enhancer generation, it struggles to fully match prior methods in conditional generation when classifier-free guidance (CFG) is applied. We hypothesize that naively applying standard CFG formulations—originally developed for continuous diffusion models—may be suboptimal for discrete flow matching. Although both Dirichlet Flow Matching and DFM apply guidance by extrapolating conditional and unconditional logits, the role of these logits differs fundamentally. In Dirichlet-based approaches, guidance softly reshapes continuous token probability distributions. In contrast, DFM applies guidance to transition rates of a discrete Markov process, influencing the timing and likelihood of discrete token transitions. For enhancer sequences, where conditioning signals are weak and global, this can lead to earlier token commitment and reduced expressiveness compared to Dirichlet Flow Matching. Notably, this issue does not arise in promoter generation, where conditioning information is incorporated directly via input concatenation rather than guidance.

In addition, distributional analyses of predicted promoter activity reveal a systematic under representation of extreme functional tails. While generated and real sequences exhibit similar medians across central quantiles, discrepancies emerge in the upper activity range. Specifically, in the 90th–100th percentile, real promoters achieve substantially higher median activity (approximately 100-fold above baseline) compared to generated promoters (approximately 60-fold). This suggests that the model has difficulty modeling rare, high-activity regulatory elements. This limitation is consistent with likelihood-based generative objectives, which emphasize global distributional coverage and tend to underweight low-volume, extreme regions of sequence space. Other possible factors for this limitation are calibration of the SEI predictor at high-activity values, or architectural limitations of the model. We leave a systematic investigation of these factors to future work. Alternative generative objectives—such as reward-driven or tail-focused modeling approaches represent a promising avenue for explicitly targeting rare, high-activity regulatory elements.

5 Conclusion

In this work, we applied Discrete Flow Matching (DFM) to DNA sequence design for both enhancer and promoter tasks. Our results show that DFM is competitive with existing state-of-the-art generative models for regulatory DNA. In particular, for conditional promoter generation conditioned on

transcriptional signal, DFM matches the performance of Fisher Flow Matching while outperforming earlier diffusion- and Dirichlet-based approaches, despite operating directly in discrete sequence space.

At the same time, our analysis highlights important limitations of likelihood-based generative models in capturing extreme functional tails, with rare high-activity regulatory elements remaining underrepresented. Our findings also suggest that the guidance mechanisms developed for continuous diffusion models do not directly translate to discrete CTMC-based flows, identifying a key barrier to effective conditional generation in discrete generative genomics. These findings suggest further research on discrete-aware guidance mechanisms, reward-augmented objectives for rare regulatory elements, and experimental validation of generated sequences.

6 Funding

The work was supported by the grant for research centers in the field of AI provided by the Ministry of Economic Development of the Russian Federation in accordance with the agreement 000000C313925P4E0002 and the agreement with HSE University no. 139-15-2025-009.

References

Zeynep Kalender Atak, Ibrahim Ihsan Taskiran, Jonas Demeulemeester, Christopher Flerin, David Mauduit, Liesbeth Minnoye, Gert Hulselmans, Valerie Christiaens, Ghanem-Elias Ghanem, Jasper Wouters, et al. Interpretation of allele-specific chromatin accessibility using cell state–aware deep learning. *Genome research*, 31(6):1082–1096, 2021.

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In *International Conference on Machine Learning*, pp. 1276–1301. PMLR, 2023.

Kathleen M. Chen, Aaron K. Wong, Olga G. Troyanskaya, and Jian Zhou. A sequence-based global map of regulatory activity for deciphering human genetics. *Nature Genetics*, 54(7):940–949, Jul 2022. doi: 10.1038/s41588-022-01102-2.

Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim. The probability flow ode is provably fast. *Advances in Neural Information Processing Systems*, 36:68552–68575, 2023.

The FANTOM Consortium, the RIKEN PMI, and CLST. A promoter-level mammalian expression atlas. *Nature*, 507(7493):462–470, 2014.

Oscar Davis, Samuel Kessler, Mircea Petrache, İsmail İ Ceylan, Michael Bronstein, and Avishek J Bose. Fisher flow matching for generative modeling over discrete data. *Advances in Neural Information Processing Systems*, 37:139054–139084, 2024.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

Anvita Gupta and James Zou. Feedback gan for dna optimizes protein functions. *Nature Machine Intelligence*, 1(2):105–111, 2019.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Chung-Chau Hon, Jordan A Ramilowski, Jayson Harshbarger, Nicolas Bertin, Owen JL Rackham, Julian Gough, Elena Denisenko, Sebastian Schmeier, Thomas M Poulsen, Jessica Severin, et al. An atlas of human long non-coding rnas with accurate 5 ends. *Nature*, 543(7644):199–204, 2017.

Jasper Janssens, Sara Aibar, Ibrahim Ihsan Taskiran, Joy N Ismail, Alicia Estacio Gomez, Gabriel Aughey, Katina I Spanier, Florian V De Rop, Carmen Bravo Gonzalez-Blas, Marc Dionne, et al. Decoding gene regulation in the fly brain. *Nature*, 601(7894):630–636, 2022.

Nathan Killoran, Leo J Lee, Andrew Delong, David Duvenaud, and Brendan J Frey. Generating and designing dna with deep generative models. *arXiv preprint arXiv:1712.06148*, 2017.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Anirban Sarkar, Yijie Kang, Nirali Somia, Pablo Mantilla, Jessica Lu Zhou, Masayuki Nagai, Ziqi Tang, Chris Zhao, and Peter Koo. Designing dna with tunable regulatory activity using score-entropy discrete diffusion. *bioRxiv*, pp. 2024–05, 2024.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to dna sequence design. *arXiv preprint arXiv:2402.05841*, 2024.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

A APPENDIX

A.0.1 DISCRETE FLOW MATCHING FORMULATION

Let $\mathcal{S} = \mathcal{T}^d$ denote the discrete space of sequences of length d, where $\mathcal{T} = [K] = \{1, \dots, K\}$ is called vocabulary. Each sequence $x = (x^1, \dots, x^d) \in \mathcal{S}$ represents a sample or a state, where $x^i \in \mathcal{T}$ is a single token. We similarly use states $y, z \in \mathcal{S}$. q is the empirical probability mass function (PMF) over these sequences, called the target distribution. We define a probability path p_t that interpolates between a source PMF p and the target PMF q. To generate a probability path p_t, we find a CTMC model $(X_t)_{0 \le t \le 1}$ defined by a learnable velocity u_t^θ. Discrete Flow Matching trains u_t^θ by minimizing the Bregman divergence that defines the loss Lipman et al. (2024).

Data and coupling The objective of discrete flow matching is to transfer samples $X_0 \sim p$ from a source PMF p to samples $X_1 \sim q$ from a target PMF q. Source and target samples can be linked by means of independent coupling $(X_0, X_1) \sim p(X_0)q(X_1)$, or connected by means of a general PMF coupling $\pi_{0,1}(x_0, x_1)$. For the task of DNA sequence generation, we consider the independent coupling where $p(x_0)$ is either a uniform probability over S giving all states equal probability, or adding a special mask token m to the vocabulary K and replacing the original tokens with mask tokens.

Discrete probability paths. Discrete Flow Matching constructs a time-dependent probability path $\{p_t\}_{t \in [0,1]}$ that interpolates between $p_0 = p$ and $p_1 = q$.

Following prior work Tong et al. (2023), we define the marginal path through a conditional mixture:

$$p_t(x) = \sum_z p_{t|Z}(x \mid z)\, p_Z(z), \tag{1}$$

where Z is an auxiliary random variable. In practice, we use *mixture discrete probability paths*, where each sequence position interpolates independently between a source token x_0^i and a target token x_1^i. For each position i,

$$p_{t|0,1}^i(x^i \mid x_0, x_1) = \kappa_t\, \delta(x^i, x_1^i) + (1 - \kappa_t)\, \delta(x^i, x_0^i), \tag{2}$$

where $\kappa(t) \in [0, 1]$ is a smooth, monotone scheduler. This construction yields a simple and interpretable discrete path in which tokens are progressively revealed over time.

The marginalization trick. Although the probability path is defined conditionally, generation requires a velocity field that depends only on the current state x_t. The marginalization trick shows that this is possible. If a conditional velocity field $u_t(\cdot, \cdot \mid Z)$ generates the conditional path $p_{t|Z}$, then the marginal velocity

$$u_t(y, x) = \mathbb{E}[u_t(y, X_t \mid Z) \mid X_t = x] \tag{3}$$

generates the marginal distribution p_t. This result allows training with conditional information (such as source–target pairs or labels) while learning a single marginal continuous-time Markov chain (CTMC) model for sampling.

Discrete Flow Matching loss. We parameterize the marginal velocity field with a neural network $u_t^\theta(y, x)$. Training proceeds by matching the model velocity to the analytically known velocity induced by the chosen probability path. The Discrete Flow Matching objective is

$$\mathcal{L}_{\mathrm{DFM}}(\theta) = \mathbb{E}_{t \sim \mathcal{U}(0,1),\, X_t \sim p_t} \left[D_{X_t}\big(u_t(\cdot, X_t), u_t^\theta(\cdot, X_t)\big) \right], \tag{4}$$

where $D_x(\cdot, \cdot)$ is a Bregman divergence over valid CTMC rate vectors. In our experiments, we use the generalized KL divergence, which is well suited for non-negative transition rates.

An equivalent conditional objective can be written using the auxiliary variable Z, and both losses yield identical learning gradients. This enables efficient training using samples from the forward process without explicitly computing marginal probabilities.

Factorized paths and velocities. Directly modeling transition rates between all possible sequences is infeasible due to the exponential size of $\mathcal{S}$. To address this, we use *factorized velocities* that allow transitions changing at most one token:

$$u_t(y, x) = \sum_{i=1}^{d} \delta(y^{\bar{i}}, x^{\bar{i}})\, u_t^i(y^i, x). \tag{5}$$

where $\bar{i} = (1, \ldots, i-1, i+1, \ldots, d)$ denotes all indices excluding i. Under this factorization, the velocity field decomposes into per-position categorical transitions, reducing the output dimensionality to $d \times K$.

Generation dynamics. At inference time, new sequences are generated by starting from a sample $X_0 \sim p$ and numerically integrating the learned velocity field from $t = 0$ to $t = 1$. The model predicts the rate of probability change of the current sample X_t in each of its N tokens. Then, each token of the sample $X_t \sim p_t$ is updated independently by

$$X_{t+h}^i = \delta(X_t^i, \cdot) + h u_t^i(\cdot, X_t), \tag{6}$$

where h is the integration step size. This procedure produces a discrete trajectory that transports samples from the source distribution to the target data distribution.

Conditional generation. For conditional sequence generation, the velocity field is augmented with auxiliary information c, such as transcription initiation signals or enhancer cell-type labels:

$$u_t^\theta(\cdot, X_t \mid c) \tag{7}$$

Classifier-free guidance (CFG) is employed to modulate the influence of conditioning. During training, conditioning information is randomly dropped, enabling the model to learn both conditional and unconditional dynamics. At inference time, the guided velocity field is computed as

$$u_{\text{guided}}^\theta(X_t) = u_t^\theta(\cdot, X_t) + \gamma \left(u_t^\theta(\cdot, X_t \mid c) - u_t^\theta(\cdot, X_t) \right), \tag{8}$$

where γ is a guidance scale controlling the strength of conditioning. In practice, CFG weighting is applied at the logit level prior to sampling, following established implementations Sun et al. (2024).