

## 729 A Proofs

### 730 A.1 Importance sampling for the generator estimation

731 This section provides a detailed derivation of the importance sampling estimator for the generator  
732 and its parametrization presented in Equation (9).

733 **Existence of densities.** Consider the measurable space  $(S, \Sigma)$  with a  $\Sigma$ -measurable reference measure  
734  $\nu$  (e.g., counting measure if  $S$  is discrete and Lebesgue measure if  $S = \mathbb{R}^d$ ), as introduced in the  
735 Section 2. Since we focus on a sampling problem, the target distribution  $p_1$  admits a  $\nu$ -density  
736  $\frac{dp_1}{d\nu} : S \rightarrow \mathbb{R}_{\geq 0}$ . Assume that the joint probability measure  $p_{t,1}$  is absolutely continuous with  
737 respect to the product measure  $\nu \otimes \nu$ . Then, all related probability measures  $p_t, p_{t|1}, p_{1|t}$  admits  
738 corresponding  $\nu$ -densities, expressed as:

$$p_t(dx_t) = \int p_{t,1}(dx_t, dx_1) \quad (26)$$

$$= \int p_{t,1}(x_t, x_1)(\nu \otimes \nu)(dx_t, dx_1) \quad (27)$$

$$= \underbrace{\left( \int p_{t,1}(x_t, x_1) \nu(dx_1) \right)}_{p_t(x_t)} \nu(dx_t), \quad (28)$$

$$p_{t,1}(dx_t, dx_1) = \int p_{t,1}(x_t, x_1)(\nu \otimes \nu)(dx_t, dx_1) \quad (29)$$

$$= \int \frac{p_{t,1}(x_t, x_1)}{p_t(x_t)} p_t(x_t) \nu(dx_t) \nu(dx_1) \quad (30)$$

$$= \int \underbrace{\frac{p_{t,1}(x_t, x_1)}{p_t(x_t)}}_{=p_{1|t}(x_1|x_t)} \nu(dx_1) p_t(dx_t), \quad (31)$$

$$p_{t,1}(dx_t, dx_1) = \int p_{t,1}(x_t, x_1)(\nu \otimes \nu)(dx_t, dx_1) \quad (32)$$

$$= \int \frac{p_{t,1}(x_t, x_1)}{p_1(x_1)} p_1(x_1) \nu(dx_1) \nu(dx_t) \quad (33)$$

$$= \int \underbrace{\frac{p_{t,1}(x_t, x_1)}{p_1(x_1)}}_{=p_{t|1}(x_t|x_1)} \nu(dx_t) p_1(dx_1), \quad (34)$$

739 where  $p_{t,1}(x_t, x_1)$  denotes the density of the joint probability measure.

740 **SNIS estimation of the generator.** We introduce a proposal distribution  $q_{1|t} : \Sigma \times S \rightarrow \mathbb{R}_{\geq 0}$ ,  
741 satisfying absolute continuity conditions:  $p_{1|t}(\cdot|x) \ll q_{1|t}(\cdot|x)$ ,  $q_{1|t}(\cdot|x) \ll \nu$  and  $\nu \ll q_{1|t}(\cdot|x)$   
742 for all  $x \in S$ . Using the marginalization trick Equation (3) and the Radon-Nikodym theorem [38,  
743 Chapter 7], we have:

$$\mathcal{L}_t f(x) = \mathbb{E}_{x_1 \sim p_{1|t}(\cdot|x)} [\mathcal{L}_{t|1}^{x_1} f(x)] \quad (35)$$

$$= \mathbb{E}_{x_1 \sim q_{1|t}(\cdot|x)} \left[ \frac{dp_{1|t}}{dq_{1|t}}(x_1|x) \mathcal{L}_{t|1}^{x_1} f(x) \right]. \quad (36)$$

744 Since the Bayes' rule holds for the  $\nu$ -density, i.e.,  $p_{1|t}(x_1|x_t) = \frac{p_{t|1}(x_t|x_1)p_1(x_1)}{p_t(x_t)}$ , we obtain:

$$\mathcal{L}_t f(x_t) = \mathbb{E}_{x_1 \sim q_{1|t}(\cdot|x_t)} \left[ \frac{dp_{1|t}}{dq_{1|t}}(x_1|x_t) \mathcal{L}_{t|1}^{x_1} f(x_t) \right] \quad (37)$$

$$= \mathbb{E}_{x_1 \sim q_{1|t}(\cdot|x_t)} \left[ \frac{dp_{1|t}}{d\nu}(x_1|x_t) \frac{d\nu}{dq_{1|t}}(x_1|x_t) \mathcal{L}_{t|1}^{x_1} f(x_t) \right] \quad (38)$$

$$= \mathbb{E}_{x_1 \sim q_{1|t}(\cdot|x_t)} \left[ \frac{p_{1|t}(x_1|x_t)}{q_{1|t}(x_1|x_t)} \mathcal{L}_{t|1}^{x_1} f(x_t) \right] \quad (39)$$

$$= \mathbb{E}_{x_1 \sim q_{1|t}(\cdot|x_t)} \left[ w(x_t, x_1) \mathcal{L}_{t|1}^{x_1} f(x_t) \right] \quad (40)$$

745 where  $w(x_t, x_1) := \frac{p_{1|t}(x_1|x_t)}{q_{1|t}(x_1|x_t)} = \frac{\tilde{p}_1(x_1)p_{t|1}(x_t|x_1)}{Z p_t(x_t) q_{1|t}(x_1|x_t)}$ . We estimate the normalization term  $Z p_t(x_t)$

746 with tractable unnormalized density  $\tilde{w}(x_t, x_1) := \frac{\tilde{p}_1(x_1)p_{t|1}(x_t|x_1)}{q_{1|t}(x_1|x_t)}$ :

$$\mathbb{E}_{x_1 \sim q_{1|t}(\cdot|x)} [\tilde{w}(x_t, x_1)] = \mathbb{E}_{x_1 \sim q_{1|t}(\cdot|x)} \left[ \frac{\tilde{p}_1(x_1)p_{t|1}(x_t|x_1)}{q_{1|t}(x_1|x_t)} \right] \quad (41)$$

$$= \int \tilde{p}_1(x_1) p_{t|1}(x_t|x_1) \nu(dx_1) \quad (42)$$

$$= \int Z \frac{\tilde{p}_1(x_1)}{Z} p_{t|1}(x_t|x_1) \nu(dx_1) \quad (43)$$

$$= Z \int p_1(x_1) p_{t|1}(x_t|x_1) \nu(dx_1) \quad (44)$$

$$= Z p_t(x_t). \quad (45)$$

747 Thus, we derive the self-normalized importance sampling (SNIS) estimator for the generator:

$$\mathcal{L}_t f(x_t) = \frac{\mathbb{E}_{x_1 \sim q_{1|t}(\cdot|x_t)} [\tilde{w}(x_t, x_1) \mathcal{L}_{t|1}^{x_1} f(x_t)]}{\mathbb{E}_{x_1 \sim q_{1|t}(\cdot|x_t)} [\tilde{w}(x_t, x_1)]} \quad (46)$$

748 **SNIS estimation of the parametrization.** Similarly, the SNIS estimator for the parameterization  $F_t$   
749 is:

$$F_t(x_t) = \mathbb{E}_{x_1 \sim p_{1|t}(\cdot|x_t)} [F_{t|1}^{x_1}(x_t)] \quad (47)$$

$$= \mathbb{E}_{x_1 \sim q_{1|t}(\cdot|x_t)} \left[ \frac{dp_{1|t}}{dq_{1|t}}(x_1|x_t) F_{t|1}^{x_1}(x_t) \right] \quad (48)$$

$$= \mathbb{E}_{x_1 \sim q_{1|t}(\cdot|x_t)} \left[ \frac{p_{1|t}(x_1|x_t)}{q_{1|t}(x_1|x_t)} F_{t|1}^{x_1}(x_t) \right] \quad (49)$$

$$= \frac{\mathbb{E}_{x_1 \sim q_{1|t}(\cdot|x_t)} [\tilde{w}(x_t, x_1) F_{t|1}^{x_1}(x_t)]}{\mathbb{E}_{x_1 \sim q_{1|t}(\cdot|x_t)} [\tilde{w}(x_t, x_1)]} \quad (50)$$

750 Specifically, the expression above suggests the Monte-Carlo (MC) estimator with  $K$  samples  
751  $x_1^{(1)}, \dots, x_1^{(K)} \sim q_{1|t}(\cdot|x)$  as follows:

$$\hat{F}_t(x_t) = \frac{\sum_{i=1}^K \tilde{w}(x_t, x_1^{(i)}) F_{t|1}^{x_1^{(i)}}(x_t)}{\sum_{i=1}^K \tilde{w}(x_t, x_1^{(i)})}. \quad (51)$$

752 This is a SNIS estimator of the parametrization  $F_t(x)$ .

## 753 A.2 Proof of Theorem 1

754 For convenience, we repeat the theorem and its assumptions below.

**Theorem 2 (Restatement of Theorem 1)** Let  $\mathcal{L}_{t|r}^{x_r}$  denote the conditional generator for conditional probability path  $p_{t|r}(\cdot|x_r)$  for  $0 \leq t \leq r \leq 1$ . If the backward transition kernels  $p_{t|r}$  satisfy the Chapman-Kolmogorov equation,

$$p_{t|1}(dx_t|x_1) = \int p_{t|r}(dx_t|x_r)p_{r|1}(dx_r|x_1), \quad (52)$$

and the conditional generators  $\mathcal{L}_{t|r}^{x_r}$  satisfy the marginal consistency as follows,

$$\mathbb{E}_{x_r \sim p_{r|1,t}(\cdot|x_1, x_t)} \left[ \mathcal{L}_{t|r}^{x_r} f(x_t) \right] = \mathcal{L}_{t|1}^{x_1} f(x_t). \quad (53)$$

Then the marginal generator can be expressed as follows, regardless of  $r$ :

$$\mathcal{L}_t f(x_t) = \mathbb{E}_{x_r \sim p_{r|t}(\cdot|x_t)} \left[ \mathcal{L}_{t|r}^{x_r} f(x_t) \right], \quad (54)$$

where  $p_{r|t}(dx_r|x)$  is the posterior distribution (i.e., the conditional distribution over intermediate state  $x_r$  given an observation  $x$  at time  $t$ ).

Define the marginal generator conditioned at time  $r$  as  $\mathcal{L}_{t;r} f(x) := \mathbb{E}_{x_r \sim p_{r|t}(\cdot|x)} [\mathcal{L}_{t|r}^{x_r} f(x)]$ . Although this definition depends explicitly on  $r$ , we aim to demonstrate its independence from  $r$ . This invariance is crucial since any dependence on  $r$  would result in conflicting objectives at times  $t < r_1, r_2$  for distinct  $r_1, r_2$ . The proof proceeds in two main steps:

1. Verify that the marginal generator  $\mathcal{L}_{t;r}$  generates the probability path  $(p_t)_{0 \leq t \leq r}$ .
2. Show the marginal generator's independence from  $r$  (i.e.,  $\mathcal{L}_{t;r} = \mathcal{L}_t$ ).

To establish the first step, it suffices to verify that the Kolmogorov Forward Equation (KFE) holds for the probability path  $(p_t)_{0 \leq t \leq r}$  and the generator  $\mathcal{L}_{t;r}$ :

$$\frac{d}{dt} \mathbb{E}_{x_t \sim p_t} [f(x_t)] = \mathbb{E}_{x_t \sim p_t} [\mathcal{L}_{t;r} f(x_t)] \quad \text{for } 0 \leq t \leq r \leq 1. \quad (55)$$

The KFE is satisfied for the conditional probability path  $p_{t|r}$  by definition:

$$\frac{d}{dt} \mathbb{E}_{x_t \sim p_{t|r}(\cdot|x_r)} [f(x_t)] = \mathbb{E}_{x_t \sim p_{t|r}(\cdot|x_r)} \left[ \mathcal{L}_{t|r}^{x_r} f(x_t) \right], \quad 0 \leq t \leq r \leq 1, x_r \in S. \quad (56)$$

Thus, we have:

$$\mathbb{E}_{x_t \sim p_t} [\mathcal{L}_{t;r} f(x_t)] = \mathbb{E}_{x_t \sim p_t} \mathbb{E}_{x_r \sim p_{r|t}(\cdot|x_t)} \left[ \mathcal{L}_{t|r}^{x_r} f(x_t) \right] \quad (57)$$

$$= \mathbb{E}_{x_r \sim p_r} \mathbb{E}_{x_t \sim p_{t|r}(\cdot|x_r)} \left[ \mathcal{L}_{t|r}^{x_r} f(x_t) \right] \quad (58)$$

$$= \mathbb{E}_{x_r \sim p_r} \frac{d}{dt} \mathbb{E}_{x_t \sim p_{t|r}(\cdot|x_r)} [f(x_t)] \quad (59)$$

$$= \frac{d}{dt} \mathbb{E}_{x_r \sim p_r} \mathbb{E}_{x_t \sim p_{t|r}(\cdot|x_r)} [f(x_t)] \quad (60)$$

$$= \frac{d}{dt} \mathbb{E}_{x_t \sim p_t} [f(x_t)] \quad (61)$$

Hence,  $\mathcal{L}_{t;r}$  indeed generates the probability path  $(p_t)_{0 \leq t \leq r}$ .

Next, we demonstrate the independence of  $\mathcal{L}_{t;r}$  from the choice of  $r$ :

$$\mathcal{L}_{t;r} f(x_t) = \mathbb{E}_{x_r \sim p_{r|t}(\cdot|x_t)} \left[ \mathcal{L}_{t|r}^{x_r} f(x_t) \right] \quad (62)$$

$$= \mathbb{E}_{x_1 \sim p_{1|t}(\cdot|x_t)} \mathbb{E}_{x_r \sim p_{r|t,1}(\cdot|x_t, x_1)} \left[ \mathcal{L}_{t|r}^{x_r} f(x_t) \right] \quad (63)$$

$$= \mathbb{E}_{x_1 \sim p_{1|t}(\cdot|x_t)} \left[ \mathcal{L}_{t|1}^{x_1} f(x_t) \right] \quad (64)$$

$$= \mathcal{L}_t f(x_t) \quad (65)$$

where the marginal consistency assumption Equation (13) is applied in the third equality. This concludes the proof.

### 776 A.3 Derivation of bootstrapping estimator for generator estimation

777 We derive a bootstrapping estimator for the marginal generator and its parametrization proposed in  
 778 Equation (21), based on the marginalization trick in Equation (54).

779 **Bootstrapped SNIS estimation of the generator.** Assume that the backward kernel  $p_{t|r}$  admits  
 780 a  $\nu$ -density. Then, the posterior  $p_{r|t}$  also admits a  $\nu$ -density. Let the proposal distribution  $q_{r|t} : \Sigma \times S \rightarrow \mathbb{R}_{\geq 0}$  satisfy  $p_{r|t}(\cdot|x) \ll q_{r|t}(\cdot|x)$ ,  $q_{r|t}(\cdot|x) \ll \nu$ , and  $\nu \ll q_{r|t}(\cdot|x)$  for all  $x \in S$ . Applying  
 782 the same change-of-measure trick as before:

$$\mathcal{L}_t f(x_t) = \mathbb{E}_{x_r \sim p_{r|t}(\cdot|x_t)} \left[ \mathcal{L}_{t|r}^{x_r} f(x_t) \right] \quad (66)$$

$$= \mathbb{E}_{x_r \sim q_{r|t}(\cdot|x_t)} \left[ \frac{dp_{r|t}}{dq_{r|t}}(x_r|x_t) \mathcal{L}_{t|r}^{x_r} f(x_t) \right] \quad (67)$$

$$= \mathbb{E}_{x_r \sim q_{r|t}(\cdot|x_t)} \left[ \frac{dp_{r|t}}{d\nu}(x_r|x_t) \frac{d\nu}{dq_{r|t}}(x_r|x_t) \mathcal{L}_{t|r}^{x_r} f(x_t) \right] \quad (68)$$

$$= \mathbb{E}_{x_r \sim q_{r|t}(\cdot|x)} \left[ \frac{p_{r|t}(x_r|x_t)}{q_{r|t}(x_r|x_t)} \mathcal{L}_{t|r}^{x_r} f(x_t) \right] \quad (69)$$

$$= \mathbb{E}_{x_r \sim q_{r|t}(\cdot|x)} \left[ \frac{p_r(x_r)p_{t|r}(x_t|x_r)}{p_t(x_t)q_{r|t}(x_r|x_t)} \mathcal{L}_{t|r}^{x_r} f(x_t) \right] \quad (70)$$

$$= \mathbb{E}_{x_r \sim q_{r|t}(\cdot|x)} \left[ w(x_t, x_r) \mathcal{L}_{t|r}^{x_r} f(x_t) \right], \quad (71)$$

783 where the importance weight  $w(x_t, x_r)$  is given by

$$w(x_t, x_r) := \frac{p_{r|t}(x_r|x_t)}{q_{r|t}(x_r|x_t)} = \frac{\tilde{p}_r(x_r)p_{t|r}(x_t|x_r)}{\tilde{p}_t(x_t)q_{r|t}(x_r|x_t)}. \quad (72)$$

784 To estimate the unnormalized density  $\tilde{p}_t(x_t)$ , we define the unnormalized importance weight:

$$\tilde{w}(x_t, x_r) := \frac{\tilde{p}_r(x_r)p_{t|r}(x_t|x_r)}{q_{r|t}(x_r|x_t)}, \quad (73)$$

785 and compute:

$$\tilde{p}_t(x_t) = \int p_{t|r}(x_t|x_r) \tilde{p}_r(x_r) \nu(dx_r) \quad (74)$$

$$= \int \frac{p_{t|r}(x_t|x_r) \tilde{p}_r(x_r)}{q_{r|t}(x_r|x_t)} q_{r|t}(x_r|x_t) \nu(dx_r) \quad (75)$$

$$= \mathbb{E}_{x_r \sim q_{r|t}(\cdot|x_t)} \left[ \frac{p_{t|r}(x_t|x_r) \tilde{p}_r(x_r)}{q_{r|t}(x_r|x_t)} \right] \quad (76)$$

$$= \mathbb{E}_{x_r \sim q_{r|t}(\cdot|x_t)} [\tilde{w}(x_t, x_r)]. \quad (77)$$

786 Thus, the marginal generator can be expressed in SNIS form as:

$$\mathcal{L}_t f(x_t) = \frac{\mathbb{E}_{x_r \sim q_{r|t}(\cdot|x_t)} [\tilde{w}(x_t, x_r) \mathcal{L}_{t|r}^{x_r} f(x_t)]}{\mathbb{E}_{x_r \sim q_{r|t}(\cdot|x_t)} [\tilde{w}(x_t, x_r)]} \quad (78)$$

787 **Bootstrapped SNIS estimation of the parametrization.** Now we derive a similar expression for  
 788 the parametrization of the generator. Suppose the conditional generator  $\mathcal{L}_{t|r}^{x_r}$  admits a parametrization  
 789  $F_{t|r}^{x_r}$  such that,

$$\mathcal{L}_{t|r}^{x_r} f(x_t) = \langle \mathcal{K}f(x_t), F_{t|r}^{x_r}(x_t) \rangle, \quad (79)$$

790 where  $\mathcal{K}$  is an operator fixed for each type of Markov processes. From the marginalization trick again:

$$\mathcal{L}_t f(x_t) = \mathbb{E}_{x_r \sim p_{r|t}(\cdot|x_t)} \left[ \mathcal{L}_{t|r}^{x_r} f(x_t) \right] \quad (80)$$

$$= \mathbb{E}_{x_r \sim p_{r|t}(\cdot|x_t)} \left[ \langle \mathcal{K} f(x_t), F_{t|r}^{x_r}(x_t) \rangle \right] \quad (81)$$

$$= \left\langle \mathcal{K} f(x_t), \underbrace{\mathbb{E}_{x_r \sim p_{r|t}(\cdot|x_t)} \left[ F_{t|r}^{x_r}(x_t) \right]}_{:= F_t(x_t)} \right\rangle \quad (82)$$

791 by linearity of the inner product. Thus, the marginal generator is parametrized by

$$F_t(x_t) = \mathbb{E}_{x_r \sim p_{r|t}(\cdot|x_t)} [F_{t|r}^{x_r}(x_t)]. \quad (83)$$

792 By applying the same importance sampling trick, we obtain the SNIS estimator:

$$F_t(x_t) = \mathbb{E}_{x_r \sim p_{r|t}(\cdot|x_t)} \left[ F_{t|r}^{x_r}(x_t) \right] \quad (84)$$

$$= \mathbb{E}_{x_r \sim q_{r|t}(\cdot|x_t)} \left[ \frac{dp_{r|t}}{dq_{r|t}}(x_r|x_t) F_{t|r}^{x_r}(x_t) \right] \quad (85)$$

$$= \mathbb{E}_{x_r \sim q_{r|t}(\cdot|x_t)} \left[ \frac{p_{r|t}(x_r|x_t)}{q_{r|t}(x_r|x_t)} F_{t|r}^{x_r}(x_t) \right] \quad (86)$$

$$= \frac{\mathbb{E}_{x_r \sim q_{r|t}(\cdot|x_t)} [\tilde{w}(x_t, x_r) F_{t|r}^{x_r}(x_t)]}{\mathbb{E}_{x_r \sim q_{r|t}(\cdot|x_t)} [\tilde{w}(x_t, x_r)]}. \quad (87)$$

793 Specifically, this yields the following Monte Carlo estimator using  $K$  samples  $x_r^{(1)}, \dots, x_r^{(K)} \sim$   
 794  $q_{r|t}(\cdot|x_t)$ :

$$\hat{F}_t(x_t) = \frac{\sum_{i=1}^K \tilde{w}(x_t, x_r^{(i)}) F_{t|r}^{x_r^{(i)}}(x_t)}{\sum_{i=1}^K \tilde{w}(x_t, x_r^{(i)})} \quad (88)$$

## B Generator estimation in the multimodal spaces

Our estimator can also be applied to the mixed state spaces  $S = X \times Y$  within the generator matching framework. Let  $\{\tilde{p}_{t|1}(\cdot|x_1)\}_{0 \leq t \leq 1}$  and  $\{\tilde{p}_{t|1}(\cdot|y_1)\}_{0 \leq t \leq 1}$  denote the conditional probability paths on the  $X$  and  $Y$ , respectively, and let  $\tilde{\mathcal{L}}_{t|1}^{x_1}$  and  $\tilde{\mathcal{L}}_{t|1}^{y_1}$  denote the corresponding conditional generators for  $x_1 \in X$  and  $y_1 \in Y$ . Assume these generators are parameterized by  $\tilde{F}_{t|1}^{x_1} : [0, 1] \times S \rightarrow V_1$  and  $\tilde{F}_{t|1}^{y_1} : [0, 1] \times S \rightarrow V_2$ , respectively. For the joint space  $S = X \times Y$ , we consider the factorized conditional path:

$$p_{t|1}(dx_t, dy_t|x_1, y_1) := \tilde{p}_{t|1}(dx_t|x_1) \tilde{p}_{t|1}(dy_t|y_1),$$

where  $x_t, x_1 \in X$  and  $y_t, y_1 \in Y$ .

According to Proposition 5 in Holderrieth et al. [16], the conditional generator associated with  $p_{t|1}$  admits the following parameterization:

$$F_{t|1}^{x_1, y_1}(x_t, y_t) = \left( \tilde{F}_{t|1}^{x_1}(x_t), \tilde{F}_{t|1}^{y_1}(y_t) \right),$$

where the sum, scalar product, and inner product are naturally defined over the tuple  $(\cdot, \cdot) \in V_1 \times V_2$ . Thus, the importance sampling estimator for the parameterized generator can be written as:

$$F_t(x_t, y_t) = \mathbb{E}_{x_1, y_1 \sim p_{1|t}(\cdot|x_t, y_t)} [F_{t|1}^{x_1, y_1}(x_t, y_t)], \quad (89)$$

$$= \mathbb{E}_{x_1, y_1 \sim q_{1|t}(\cdot|x_t, y_t)} \left[ \frac{dp_{1|t}}{dq_{1|t}}(x_1, y_1|x_t, y_t) \left( \tilde{F}_{t|1}^{x_1}(x_t), \tilde{F}_{t|1}^{y_1}(y_t) \right) \right]. \quad (90)$$

As in the uni-modality case, this leads to a self-normalized importance sampling estimator, which can be directly extended to the bootstrapping setting. This demonstrates the generality and flexibility of our framework in handling multi-modal spaces.

## C Example of EGM with application to flow and masked diffusion

### C.1 Generator of flow and jump model

In this section, we provide the definition of flow and discrete jump models, their generators and parametrizations. For the case of diffusion processes or more rigorous derivations, we refer the reader to Holderrieth et al. [16]. The discrete jump model is often referred to as a continuous-time Markov chain (CTMC).

**Flow model.** Let the state space be  $S = \mathbb{R}^d$ , and let  $u_t : \mathbb{R}^d \times [0, 1] \rightarrow \mathbb{R}^d$  be a time-dependent vector field. The flow  $X_t$  is defined by the following ordinary differential equation:

$$\frac{dX_t}{dt} = u_t(X_t), \quad X_0 \sim p_0. \quad (91)$$

By definition of the generator, the generator of the flow model is given by

$$\mathcal{L}_t f(x) = \lim_{h \rightarrow 0} \frac{\mathbb{E}[f(X_{t+h}) | X_t = x] - f(x)}{h} \quad (92)$$

$$= \lim_{h \rightarrow 0} \frac{\mathbb{E}[f(X_t + hu_t(X_t) + o(h)) | X_t = x] - f(x)}{h} \quad (93)$$

$$= \lim_{h \rightarrow 0} \frac{\mathbb{E}[f(X_t) + h\nabla f(x)^T u_t(X_t) + o(h) | X_t = x] - f(x)}{h} \quad (94)$$

$$= \nabla f(x)^T u_t(X_t), \quad (95)$$

where we use a first-order Taylor expansion. Hence, the generator of the flow model admits the following linear parametrization:

$$\mathcal{L}_t f(x) = \langle \mathcal{K}f(x), u_t(x) \rangle, \quad \mathcal{K}f(x) = \nabla f(x), \quad (96)$$

i.e., the generator is parameterized by the ODE vector field  $u_t$ , and EGM aims to learn  $u_t$  via its conditional counterpart  $u_{t|1}$ .

**Discrete jump model.** Let the state space  $S$  be discrete with  $|S| < \infty$ , and define the time-dependent transition rate matrix  $Q_t : S \times S \times [0, 1] \rightarrow \mathbb{R}$  such that  $Q_t(x, x) = -\sum_{y \neq x} Q_t(y, x)$  and  $Q_t(y, x) \geq 0$  for all  $y \neq x$ . The CTMC is defined by the transition rule:

$$X_{t+h} \sim p_{t+h|t}(\cdot | X_t) = \delta_{X_t}(\cdot) + hQ_t(\cdot, X_t). \quad (97)$$

We derive the generator informally; see Davis [39] for a formal treatment:

$$\mathcal{L}_t f(x) = \lim_{h \rightarrow 0} \frac{\mathbb{E}[f(X_{t+h}) | X_t = x] - f(x)}{h} \quad (98)$$

$$= \lim_{h \rightarrow 0} \frac{\mathbb{E}[f(X_{t+h}) - f(X_t) | X_t = x, \text{Jump in } [t, t+h)] \mathbb{P}(\text{Jump in } [t, t+h))}{h} \quad (99)$$

$$+ \lim_{h \rightarrow 0} \underbrace{\frac{\mathbb{E}[f(X_{t+h}) - f(X_t) | X_t = x, \text{No jump in } [t, t+h)] \mathbb{P}(\text{No jump in } [t, t+h))}{h}}_{=0} \quad (100)$$

$$= \lim_{h \rightarrow 0} \frac{\sum_{y \neq x} (f(y) - f(x)) \left( \frac{Q_t(y, x)h}{-Q_t(x, x)h} \right) (-Q_t(x, x)h)}{h} \quad (101)$$

$$= \sum_{y \neq x} (f(y) - f(x)) Q_t(y, x) = \sum_{y \in S} f(y) Q_t(y, x) \quad (102)$$

Therefore, the generator of the CTMC can be linearly parameterized as:

$$\mathcal{L}_t f(x) = \langle \mathcal{K}f(x), Q_t(\cdot, x) \rangle, \quad \mathcal{K}f(x) = (f(y) - f(x))_{y \in S}, \quad \langle a, b \rangle := \sum_{y \in S} a_y b_y, \quad (103)$$

i.e., the generator is parameterized by the transition rate matrix  $Q_t(\cdot, x)$ , and EGM aims to learn  $Q_t$  via its conditional form  $Q_{t|1}$ .

**Remark on linear parametrization.** Under mild regularity conditions (e.g., Feller processes), Holderrieth et al. [16] shows that Markov processes on both discrete and continuous state spaces can be universally expressed via linear parameterizations:

1. **Discrete state space** ( $|S| < \infty$ ): The generator is parameterized by the transition rate matrix  $Q_t$ , corresponding to a CTMC.
2. **Euclidean space** ( $S = \mathbb{R}^d$ ): The generator is parameterized as a combination of flow, diffusion, and jump components.

This implies that, like GM, EGM is capable of modeling a wide range of Markov processes on both discrete and Euclidean spaces.

## C.2 Application to the conditional OT flow model

This section details the application of the EGM framework to flow models defined via the conditional optimal transport (CondOT) path.

**Definition of the CondOT path.** The conditional OT probability path is defined as:

$$X_t = tX_1 + (1-t)X_0, \quad (104)$$

where  $X_1 \sim p_1$ ,  $X_0 \sim p_0 = \mathcal{N}(0, I)$ , and  $X_0, X_1$  are independent. It linearly interpolates between a Gaussian prior and the target distribution. By construction, the conditional distribution is given by:

$$p_{t|1}(x_t|x_1) = \mathcal{N}(x_t; tx_1, (1-t)^2 I). \quad (105)$$

**EGM on the CondOT path.** First, consider a naive implementation of EGM with a simple proposal distribution defined as:

$$q_{1|t}(x_1|x_t) \propto p_{t|1}(x_t|x_1) = \mathcal{N}(x_t; tx_1, (1-t)^2 I) \quad (106)$$

$$\propto \exp\left(-\frac{\|x_t - tx_1\|_2^2}{2(1-t)^2}\right) \quad (107)$$

$$= \exp\left(-\frac{\|x_1 - \frac{x_t}{t}\|_2^2}{2\frac{(1-t)^2}{t^2}}\right), \quad (108)$$

which implies that

$$q_{1|t}(x_1|x_t) = \mathcal{N}\left(x_1; \frac{x_t}{t}, \frac{(1-t)^2}{t^2} I\right). \quad (109)$$

This choice yields a simple importance weight of the form  $\tilde{w}(x_t, x_1) = \tilde{p}_1(x_1)/Z_{1|t}(x_t)$ .

Using the identity from Equation (51), the estimated vector field  $u_t(x_t)$  becomes:

$$u_t(x_t) = \frac{\sum_i \tilde{p}_1(x_1^{(i)}) u_{t|1}^{x_1^{(i)}}(x_t)}{\sum_i \tilde{p}_1(x_1^{(i)})} \quad (110)$$

$$= \frac{\sum_i \tilde{p}_1(x_1^{(i)}) u_{t|1}^{x_1^{(i)}}(x_t)}{\sum_i \tilde{p}_1(x_1^{(i)})}, \quad (111)$$

where  $x_1^{(i)} \sim q_{1|t}(\cdot|x_t)$ . This is precisely the same estimator used in Woo and Ahn [34] for the flow-based sampler.

**Assumption check for bootstrapping.** Next, we derive the bootstrapping estimator. We construct the backward transition kernel  $p_{t|r}$  satisfying the marginal consistency Equation (52):

$$p_{t|r}(x_t|x_r) = \mathcal{N}(x_t; \frac{t}{r}x_r, \sigma_t I), \quad \sigma_t = (1-t)^2 - \frac{t^2}{r^2}(1-r)^2. \quad (112)$$

We verify the consistency via:

$$\int p_{t|r}(x_t|x_r) p_{r|1}(x_r|x_1) dx_r = p_{t|1}(x_t|x_1). \quad (113)$$



855 Using reparameterization tricks  $X_t = \frac{t}{r}X_r + \sqrt{\sigma_t}\epsilon_t$ ,  $X_r = rX_1 + (1-r)\epsilon_r$ ,  $\epsilon_t \perp \epsilon_r$ , we have:

$$X_t = \frac{t}{r}(rX_1 + (1-r)\epsilon_r) + \sqrt{\sigma_t}\epsilon_t \quad (114)$$

$$= tX_1 + \frac{t}{r}(1-r)\epsilon_r + \sqrt{\sigma_t}\epsilon_t \quad (115)$$

$$\stackrel{d}{=} tX_1 + (1-t)\epsilon'_t, \quad \epsilon'_t \sim \mathcal{N}(0, I), \quad (116)$$

856 where  $\stackrel{d}{=}$  denotes that two random variables have same distribution. Thus, marginal consistency holds.

857 The conditional vector field  $u_{t|r}$  is defined as:

$$u_{t|r}(x_t|x_r) = \frac{1}{r}x_r + \frac{\dot{\sigma}_t}{2\sigma_t}(x_t - \frac{t}{r}x_r). \quad (117)$$

858 This vector field arises naturally from differentiation of the reparameterization:

$$X_t = \frac{t}{r}X_r + \sqrt{\sigma_t}X_0 \implies \dot{X}_t = \frac{1}{r}X_r + \sqrt{\dot{\sigma}_t}X_0 \quad (118)$$

$$= \frac{1}{r}X_r + \frac{\sqrt{\dot{\sigma}_t}}{\sqrt{\sigma_t}}\left(X_t - \frac{t}{r}X_r\right) \quad (119)$$

$$= \frac{1}{r}X_r + \frac{\dot{\sigma}_t}{2\sigma_t}\left(X_t - \frac{t}{r}X_r\right). \quad (120)$$

859 Now, verify that the conditional vector field  $u_{t|r}$  satisfies the marginal consistency [Equation \(53\)](#):

$$\mathbb{E}_{x_r \sim p_{r|1,t}(\cdot|x_1, x_t)}[u_{t|r}(x_t|x_r)] = u_{t|1}(x_t|x_1). \quad (121)$$

860 With  $p_{r|1,t}(x_r|x_1, x_t) = \frac{p_{t|r}(x_t|x_r)p_{r|1}(x_r|x_1)}{p_{t|1}(x_t|x_1)}$  being Gaussian, its mean is explicitly:

$$\mu_{r|1,t}(x_1, x_t) = \frac{t(1-r)^2}{r(1-t)^2}x_t + \frac{r\sigma_t}{(1-t)^2}x_1. \quad (122)$$

861 Direct calculation confirms consistency:

$$\mathbb{E}_{x_r \sim p_{r|1,t}(\cdot|x_1, x_t)}[u_{t|r}(x_t|x_r)] = \mathbb{E}_{x_r \sim p_{r|1,t}(\cdot|x_1, x_t)}\left[\frac{1}{r}x_r + \frac{\dot{\sigma}_t}{2\sigma_t}\left(x_t - \frac{t}{r}x_r\right)\right] \quad (123)$$

$$= \frac{1}{r}\mathbb{E}_{x_r \sim p_{r|1,t}(\cdot|x_1, x_t)}[x_r] + \frac{\dot{\sigma}_t}{2\sigma_t}\left(x_t - \frac{t}{r}\mathbb{E}_{x_r \sim p_{r|1,t}(\cdot|x_1, x_t)}[x_r]\right) \quad (124)$$

$$= \frac{1}{r}\mu_{r|1,t}(x_1, x_t) + \frac{\dot{\sigma}_t}{2\sigma_t}\left(x_t - \frac{t}{r}\mu_{r|1,t}(x_1, x_t)\right). \quad (125)$$

862 The first term  $\frac{1}{r}\mu_{r|1,t}$  reduces to,

$$\frac{1}{r}\mu_{r|1,t}(x_1, x_t) = \frac{t(1-r)^2}{r^2(1-t)^2}x_t + \frac{\sigma_t}{(1-t)^2}x_1 \quad (126)$$

$$= \frac{t(1-r)^2x_t + r^2\sigma_t x_1}{r^2(1-t)^2} \quad (127)$$

$$= \frac{t(1-r)^2}{r^2(1-t)^2}(x_t - tx_1) + x_1, \quad (128)$$

863 where we used  $\sigma_t = (1-t)^2 - \frac{t^2}{r^2}(1-r)^2$  in the third equality.

864 The part of second term  $x_t - \frac{t}{r}\mu_{r|1,t}(x_1, x_t)$  reduces to,

$$x_t - \frac{t}{r}\mu_{r|1,t}(x_1, x_t) = x_t - \frac{t^2(1-r)^2}{r^2(1-t)^2}x_t - \frac{t\sigma_t}{(1-t)^2}x_1 \quad (129)$$

$$= \frac{r^2(1-t)^2 - t^2(1-r)^2}{r^2(1-t)^2}x_t - \frac{t\sigma_t}{(1-t)^2}x_1 \quad (130)$$

$$= \frac{\sigma_t}{(1-t)^2}x_t - \frac{t\sigma_t}{(1-t)^2}x_1, \quad (131)$$

865 where we used  $\sigma_t = (1-t)^2 - \frac{t^2}{r^2}(1-r)^2$  in the third equality.

866 Put it all together, we conclude that,

$$\mathbb{E}_{x_r \sim p_{r|1,t}(\cdot|x_1, x_t)}[u_{t|r}(x_t|x_r)] = \frac{1}{r}\mu_{r|1,t}(x_1, x_t) + \frac{\dot{\sigma}_t}{2\sigma_t}(x_t - \frac{t}{r}\mu_{r|1,t}(x_1, x_t)) \quad (132)$$

$$= \frac{t(1-r)^2}{r^2(1-t)^2}(x_t - tx_1) + x_1 + \frac{\dot{\sigma}_t}{2\sigma_t} \left( \frac{\sigma_t}{(1-t)^2}x_t - \frac{t\sigma_t}{(1-t)^2}x_1 \right) \quad (133)$$

$$= \frac{t(1-r)^2}{r^2(1-t)^2}(x_t - tx_1) + x_1 + \frac{\dot{\sigma}_t}{2(1-t)^2}(x_t - tx_1) \quad (134)$$

$$= \left( \frac{t(1-r)^2}{r^2} + \frac{\dot{\sigma}_t}{2} \right) \frac{x_t - tx_1}{(1-t)^2} + x_1 \quad (135)$$

$$= (1-t) \frac{x_t - tx_1}{(1-t)^2} + x_1 \quad (136)$$

$$= \frac{x_1 - x_t}{1-t} \quad (137)$$

$$= u_{t|1}(x_t|x_1), \quad (138)$$

867 which implies the proposed transition kernel  $p_{t|r}(x_t|x_r)$  and conditional vector field  $u_{t|r}(x_t|x_r)$   
868 satisfies the assumption of our [Theorem 1](#).

869 **Bootstrapped estimator for the CondOT.** Lastly, we define the bootstrapping estimator for the  
870 CondOT flow model using proposal as follows:

$$q_{r|t}(x_r|x_t) \propto p_{t|r}(x_t|x_r) = \mathcal{N}(x_t; \frac{t}{r}x_r, \sigma_t I) \quad (139)$$

$$\propto \exp \left( -\frac{\|x_t - \frac{t}{r}x_r\|_2^2}{2\sigma_t} \right) \quad (140)$$

$$\propto \exp \left( -\frac{\|x_r - \frac{r}{t}x_t\|_2^2}{2\frac{r^2}{t^2}\sigma_t} \right), \quad (141)$$

871 which implies that

$$q_{r|t}(x_r|x_t) = \mathcal{N} \left( x_r; \frac{r}{t}x_t, \frac{r^2}{t^2}\sigma_t I \right). \quad (142)$$

872 The bootstrapping estimator is then given by:

$$\hat{u}_t(x_t) = \frac{\sum_{i=1}^K \tilde{w}(x_t, x_r^{(i)}) u_{t|r}(x_t|x_r)}{\sum_{i=1}^K \tilde{w}(x_t, x_r^{(i)})}, \quad \tilde{w}(x_t, x_r) = \tilde{p}_r(x_r) = \exp(-\mathcal{E}_r^\phi(x_r)), \quad (143)$$

873 where samples  $x_r^{(1)}, \dots, x_r^{(K)} \sim q_{r|t}(\cdot|x_t)$  and  $\mathcal{E}_r^\phi(x_r)$  is learned energy estimator.

### 874 C.3 Application to the masked diffusion model

875 This section describes how the EGM framework can be applied to discrete jump models using the  
876 masked diffusion path.

877 **Definition of masked diffusion path.** We define the masked diffusion path as follows:

$$p_{t|r}(x_t|x_r) = \frac{\kappa_t}{\kappa_r} \delta_{x_r}(x_t) + \left( 1 - \frac{\kappa_t}{\kappa_r} \right) \delta_M(x_t). \quad (144)$$

878 where  $\kappa_t : [0, 1] \rightarrow \mathbb{R}_{>0}$  is an increasing function satisfying  $\kappa_0 = 0, \kappa_1 = 1$ ,  $M$  is the mask token,  
879 and  $\delta_x$  is the Dirac-delta distribution centered at  $x$ . Next, we derive the conditional transition rate  
880 matrix generating the conditional probability path  $p_{t|r}(\cdot|x_r)$ . Starting from the Kolmogorov forward

equation, we have:

$$\frac{d}{dt}p_{t|r}(y_t|x_r) = \frac{\dot{\kappa}_t}{\kappa_r}(\delta_{x_r}(y_t) - \delta_M(y_t)) \quad (145)$$

$$= \frac{\dot{\kappa}_t}{\kappa_r} \frac{1}{\kappa_r - \kappa_t}((\kappa_r - \kappa_t)\delta_{x_r}(y_t) - (\kappa_r - \kappa_t)\delta_M(y_t)) \quad (146)$$

$$= \frac{\dot{\kappa}_t}{\kappa_r} \frac{\kappa_r}{\kappa_r - \kappa_t}(\delta_{x_r}(y_t) - p_{t|r}(y_t|x_r)) \quad (147)$$

$$= \sum_{x_t} \frac{\dot{\kappa}_t}{\kappa_r - \kappa_t}(\delta_{x_r}(y_t) - \delta_{x_t}(y_t))p_{t|r}(x_t|x_r) \quad (148)$$

$$= \sum_{x_t} u_{t|r}(y_t, x_t|x_r)p_{t|r}(x_t|x_r), \quad (149)$$

thus obtaining  $u_{t|r}(y_t, x_t|x_r) = \frac{\dot{\kappa}_t}{\kappa_r - \kappa_t}(\delta_{x_r}(y_t) - \delta_{x_t}(y_t))$ .

**EGM on the masked diffusion path.** We first introduce a naive implementation of EGM using a simple proposal distribution defined as:

$$q_{1|t}(x_1|x_t) \propto p_{t|1}(x_t|x_1) = \kappa_t\delta_{x_1}(x_t) + (1 - \kappa_t)\delta_M(x_t) \quad (150)$$

$$(151)$$

which implies:

$$q_{1|t}(x_1|x_t) = \begin{cases} \text{Unif}(x; S - M) & (x = M) \\ \delta_{x_t}(x_1) & (x \neq M) \end{cases}. \quad (152)$$

This yields the simple importance weight  $\tilde{w}(x_t, x_1) = \tilde{p}_1(x_1)/Z_{1|t}(x_t)$ . Following Equation (51), the estimator for the transition matrix  $u_t(y_t, x_t)$  becomes:

$$\hat{u}_t(y_t, x_t) = \frac{\sum_{i=1}^K \tilde{p}_1(x_1)u_{t|1}(y_t, x_t|x_1^{(i)})}{\sum_{i=1}^K \tilde{p}_1(x_1^{(i)})} \quad (153)$$

where samples  $x_1^{(1)}, \dots, x_1^{(K)} \sim q_{1|t}(\cdot|x_t)$ .

In practice, the state space  $S = [N]^D$  factorizes along dimensions, where  $D$  is sequence length and  $[N] = \{1, \dots, N\}$ . We thus factorize the masked diffusion path as follows:

$$p_{t|1}(x_t|x_1) = \prod_{i=1}^D p_{t|1}^i(x_t^i|x_1^i), \quad p_{t|1}^i(x_t^i|x_1^i) = \kappa_t\delta_{x_1^i}(x_t^i) + (1 - \kappa_t)\delta_M(x_t^i) \quad (154)$$

where  $x^i \in [N]$  denotes the  $i$ -th token of the sequence  $x \in S$ . The proposal  $q_{1|t}$  and the transition rate matrix  $u_t(y, x)$  factorize accordingly. The proposal factorizes as:

$$q_{1|t}(x_1|x_t) = \prod_{i=1}^D q_{1|t}^i(x_1^i|x_t^i) \propto \prod_{i=1}^D p_{t|1}^i(x_t^i|x_1^i), \quad (155)$$

where  $q_{1|t}^i$  is a proposal defined over each dimensions. The transition matrix factorizes as:

$$u_t(y, x) = \sum_{i=1}^D \delta(y^{-i}, x^{-i})u_t^i(y^i, x), \quad (156)$$

where  $x^{-i}$  denotes the  $x$  without  $i$ -th token and  $u_t^i$  is transition rate for each dimension. Hence, our neural network is trained to predict the  $D \times N$  matrix  $\text{NN}_\theta : (x_t, t) \mapsto (u_t^i(y_t^i, x_t^i))_{1 \leq i \leq D, y_t^i \in [N]}$ .

**Assumption check for bootstrapping.** The backward transition kernel  $p_{t|r}$  of masked diffusion satisfies the marginal consistency since it defines the Markov process (noising process of masked diffusion). Thus, it is suffice to show that the conditional transition rate matrix  $u_{t|r}(y_t, x_t|x_r)$  satisfies the marginal consistency Equation (53):

$$\mathbb{E}_{x_r \sim p_{r|1,t}(\cdot|x_1, x_t)}[u_{t|r}(y_t, x_t|x_r)] = u_{t|1}(y_t, x_t|x_1). \quad (157)$$

900 This condition can be confirmed via explicit calculations. Note that  $p_{r|1,t}(x_r|x_1, x_t) =$   
 901  $\frac{p_{t|r}(x_t|x_r)p_{r|1}(x_r|x_1)}{p_{t|1}(x_t|x_1)}.$

$$(L.H.S.) = \sum_{x_r} \frac{\dot{\kappa}_t}{\kappa_r - \kappa_t} (\delta_{x_r}(y_t) - \delta_{x_t}(y_t)) p_{r|1,t}(x_r|x_1, x_t) \quad (158)$$

$$= \sum_{x_r} \frac{\dot{\kappa}_t}{\kappa_r - \kappa_t} (\delta_{x_r}(y_t) - \delta_{x_t}(y_t)) \frac{p_{t|r}(x_t|x_r)p_{r|1}(x_r|x_1)}{p_{t|1}(x_t|x_1)} \quad (159)$$

$$= \frac{\dot{\kappa}_t}{(\kappa_r - \kappa_t)p_{t|1}(x_t|x_1)} \sum_{x_r} (\delta_{x_r}(y_t) - \delta_{x_t}(y_t)) p_{t|r}(x_t|x_r)p_{r|1}(x_r|x_1) \quad (160)$$

$$= \frac{\dot{\kappa}_t}{(\kappa_r - \kappa_t)p_{t|1}(x_t|x_1)} \left( (\delta_M(y_t) - \delta_{x_t}(y_t)) p_{t|r}(x_t|M)p_{r|1}(M|x_1) \right. \quad (161)$$

$$\left. + (\delta_{x_1}(y_t) - \delta_{x_t}(y_t)) p_{t|r}(x_t|x_1)p_{r|1}(x_1|x_1) \right) \quad (162)$$

$$= \frac{\dot{\kappa}_t}{(\kappa_r - \kappa_t)p_{t|1}(x_t|x_1)} \left( (\delta_M(y_t) - \delta_{x_t}(y_t)) \delta_M(x_t)(1 - \kappa_r) \right. \quad (163)$$

$$\left. + (\delta_{x_1}(y_t) - \delta_{x_t}(y_t)) p_{t|r}(x_t|x_1)\kappa_r \right) \quad (164)$$

$$= \begin{cases} 0 & (x_t = x_1) \\ \frac{\dot{\kappa}_t}{(\kappa_r - \kappa_t)(1 - \kappa_t)} (\delta_{x_1}(y_t) - \delta_{x_t}(y_t)) \left(1 - \frac{\kappa_t}{\kappa_r}\right) \kappa_r & (x_t = M) \end{cases} \quad (165)$$

$$= \frac{\dot{\kappa}_t}{1 - \kappa_t} (\delta_{x_1}(y_t) - \delta_{x_t}(y_t)) \quad (166)$$

$$= u_{t|1}(y_t, x_t|x_1) \quad (167)$$

902 where we used the fact that  $p_{r|1}(x_r|x_1) > 0$  for only  $x_r = M$  or  $x_r = x_1$  in the fourth equality.  
 903 Hence, the proposed transition kernel  $p_{t|r}$  and conditional transition rate matrix  $u_{t|r}$  satisfies the  
 904 assumption of our [Theorem 1](#).

905 **Bootstrapped estimator for masked diffusion.** Lastly, we define the bootstrapping estimator for the  
 906 transition rate matrix of masked diffusion model. We use the following proposal:

$$q_{r|t}(x_r|x_t) \propto p_{t|r}(x_t|x_r) = \frac{\kappa_t}{\kappa_r} \delta_{x_r}(x_t) + \left(1 - \frac{\kappa_t}{\kappa_r}\right) \delta_M(x_t) \quad (168)$$

$$= \begin{cases} \frac{\kappa_t}{\kappa_r} & (x_t \neq M, x_r = x_t) \\ 0 & (x_t \neq M, x_r \neq x_t) \\ 1 - \frac{\kappa_t}{\kappa_r} & (x_t = M, x_r \neq M) \\ 1 & (x_t = M, x_r = M) \end{cases} \quad (169)$$

907 which implies,

$$q_{r|t}(x_r|x_t) = \begin{cases} \delta_{x_t}(x_r) & (x_t \neq M) \\ \text{Cat}(1 - \frac{\kappa_t}{\kappa_r}, \dots, 1 - \frac{\kappa_t}{\kappa_r}, 1) & (x_t = M) \end{cases} \quad (170)$$

908 where the mask token is the last token  $M = N$  and Cat is the categorical distribution with unnormal-  
 909 ized weight.

910 The bootstrapping estimator is given by:

$$\hat{u}_t(y_t, x_t) = \frac{\sum_{i=1}^K \tilde{w}(x_t, x_r^{(i)}) u_{t|r}(y_t, x_t|x_r^{(i)})}{\sum_{i=1}^K \tilde{w}(x_t, x_r^{(i)})}, \quad \tilde{w}(x_t, x_r) = \tilde{p}_r(x_r) = \exp(-\mathcal{E}_r^\phi(x_r)), \quad (171)$$

911 where samples  $x_r^{(1)}, \dots, x_r^{(K)} \sim q_{r|t}(\cdot|x_t)$  and  $\mathcal{E}_r^\phi(x_r)$  is learned energy estimator.

912 **Learning energy with generalized NEM objective.** We train the  $\mathcal{E}_r^\phi$  with the following estimator  
 913 for the intermediate energy:

$$\mathcal{E}_r(x_r) = -\log \mathbb{E}_{x_1 \sim q_{1|r}(\cdot|x_r)} [\exp(-\mathcal{E}_1(x_1))] - \log Z_{1|r}(x_r). \quad (172)$$

914 For masked diffusion, the partition function  $Z_{1|r}(x_r)$  explicitly depends on  $x_r$ , is given by:

$$Z_{1|r}(x_r) = \sum_{x_1} p_{r|1}(x_r|x_1) \quad (173)$$

$$= \sum_{x_1} \kappa_r \delta_{x_1}(x_r) + (1 - \kappa_r) \delta_M(x_r) \quad (174)$$

$$= \begin{cases} (N-1)(1 - \kappa_r) & (x_r = M) \\ \kappa_r & (x_r \neq M) \end{cases} \quad (175)$$

915 where  $N$  is the number of token in the state space  $S = [N]^D$ .

## D Additional details on the experiments

In this section, we provide detailed descriptions of the experimental tasks, evaluation metrics, and experimental setups used throughout this work. The code is available at [here](#).

### D.1 Task details

**Discrete Ising model.** We consider the Ising model defined on a 2D grid  $\{-1, 1\}^{L \times L}$  with size  $L$ . The energy function  $\mathcal{E} : \{-1, 1\}^{L \times L} \rightarrow \mathbb{R}$  is given by:

$$\mathcal{E}(x) = \beta \left( -J \sum_{\langle i, j \rangle} x_i x_j + \mu \sum_i x_i \right), \quad (176)$$

where  $\langle i, j \rangle$  denotes pairs of neighboring spins,  $J$  is the interaction strength,  $\mu$  is the magnetic moment, and  $\beta$  is the inverse temperature. We employ periodic boundary conditions and specifically focus on the ferromagnetic setting ( $J > 0$ ) without external fields ( $\mu = 0$ ), reducing the energy function to:

$$\mathcal{E}(x) = -\beta J \sum_{\langle i, j \rangle} x_i x_j. \quad (177)$$

We fix the interaction strength at  $J = 1.0$  and examine various temperatures through  $\beta$ .

For evaluation, approximate ground truth samples are generated using an extended Gibbs sampling run consisting of 10k burn-in steps, thinning every 10 steps, and 4 parallel chains, collecting 300k samples in total.

**GB-RBM.** The Gaussian-Bernoulli Restricted Boltzmann Machine (GB-RBM) task involves two continuous visible units following Gaussian distributions and three binary hidden units following Bernoulli distributions, with the energy function:

$$\mathcal{E}(x_1, x_2) = \Sigma^{-1} \|x_1 - a\|_2^2 - \langle b, x_2 \rangle - \Sigma^{-1} x_1^T W x_2, \quad (178)$$

where  $x_1, a \in \mathbb{R}^2$ ,  $x_2, b \in \{0, 1\}^3$ ,  $\Sigma \in \mathbb{R}$ , and  $W \in \mathbb{R}^{2 \times 3}$ . Parameters are selected to induce multiple modes, specifically six modes in continuous dimensions (see [Figure 5](#)). We set:

$$a = [0, 0], \quad b = [-5, -5, -5], \quad \Sigma = 2, \quad W = \begin{pmatrix} 10 & 0 & 10 \\ 0 & 10 & 0 \end{pmatrix}. \quad (179)$$

Approximately ground truth samples are generated using Gibbs sampling with 10k burn-in steps, 100-step thinning intervals, and 100 parallel chains, collecting 100k samples.

**JointDW4.** JointDW4 exemplifies the molecular sequence-structure co-generation task, extending the classical four-particle double-well (DW4) benchmark with particle-type-dependent interactions. This setup includes 4 particles in 2D space, each assigned discrete types, yielding a 12-dimensional (8 continuous, 4 discrete) energy function:

$$\mathcal{E}_{\text{JointDW4}}(x, t) = \frac{1}{2\tau} \sum_{i, j} a(t_i, t_j)(d_{ij} - d_0) + b(t_i, t_j)(d_{ij} - d_0)^2 + c(t_i, t_j)(d_{ij} - d_0)^4, \quad (180)$$

where  $d_{ij} = \|x_i - x_j\|_2$  is a Euclidean distance between the particle  $i, j$  and  $t_i \in \{1, 2\}$  is the type of particle  $i$ . The parameters are set as follows:

$$a(\cdot, \cdot) = 0, \quad \tau = 1, \quad d_0 = 2, \quad b = \begin{pmatrix} -3.0 & -2.5 \\ -2.5 & -2.8 \end{pmatrix}, \quad c = \begin{pmatrix} 0.8 & 0.4 \\ 0.4 & 0.6 \end{pmatrix}, \quad (181)$$

where  $b(t_i, t_j) = b_{t_i t_j}$  and  $c(t_i, t_j) = c_{t_i t_j}$ .

Ground truth samples are similarly obtained from Gibbs sampling, running 10k burn-in steps, thinning every 50 steps, across 100 parallel chains, collecting 100k samples in total.

**JointMoG.** The JointMoG extends a Gaussian mixture benchmark commonly used for evaluating diffusion samplers. It includes one continuous dimension  $x \in \mathbb{R}$  and one binary dimension  $b \in \{-1, 1\}$ :

$$\mathcal{E}_{\text{2D-JointMoG}}(x, b) = \frac{1}{2\sigma^2} \|x - b\|_2^2, \quad (182)$$

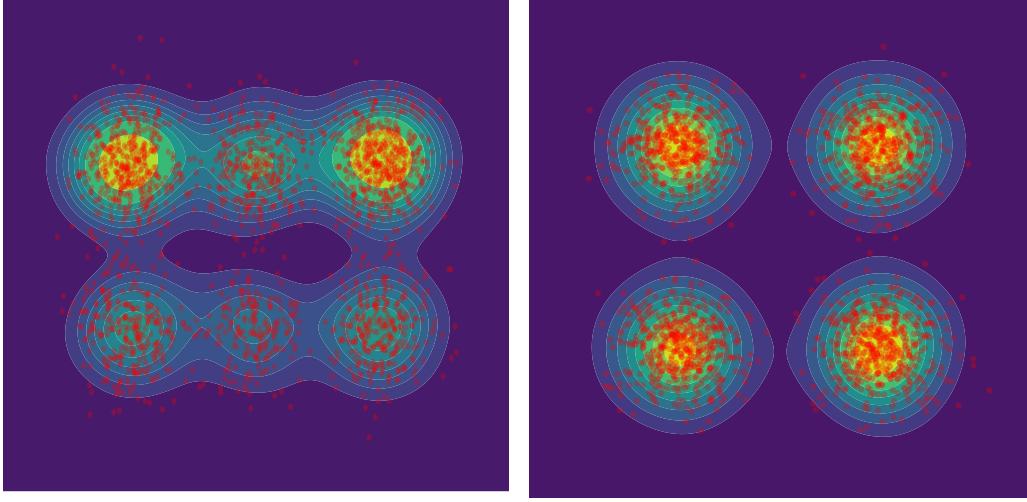


Figure 5: Ground truth sample plot of GB-RBM (left) and JointMoG (right). Samples are projected onto the first two continuous dimensions.

with standard deviation  $\sigma$ . We scale this model to 20 dimensions (10 continuous, 10 discrete) for benchmarking:

$$\mathcal{E}_{\text{JointMoG}}(x, b) = \sum_i \frac{1}{2\sigma^2} \|x_i - b_i\|_2^2, \quad (183)$$

with  $\sigma = 0.3$  to create clearly separated modes. Exact sampling is possible by first sampling discrete variables uniformly and subsequently sampling continuous variables from corresponding Gaussians, providing exact evaluation samples.

## D.2 Metrics

Evaluation metrics in our experiments primarily utilize Wasserstein distances, computed via the Python Optimal Transport (POT) library [40] using exact linear programming. Specifically, we measure the distances between 2000 empirical samples generated by our samplers and 2000 ground truth samples uniformly selected from extensive Gibbs sampling or exact sampling processes.

The Wasserstein distance of order  $p$  between two probability measures  $\mu$  and  $\nu$  is defined as:

$$\mathcal{W}_p(\mu, \nu) = \left( \inf_{\pi \in \Pi(\mu, \nu)} \int d(x, y)^p d\pi(x, y) \right)^{1/p}, \quad (184)$$

where  $\Pi(\mu, \nu) = \{\pi \in \mathcal{P}(X \times X) \mid \pi(A \times X) = \mu(A), \pi(X \times B) = \nu(B)\}$  is the set of all couplings between  $\mu$  and  $\nu$ , and  $d(x, y)$  denotes the metric on the space.

**Energy 1-Wasserstein ( $\mathcal{E}\text{-}\mathcal{W}_1$ ).** We use the Energy 1-Wasserstein distance as our primary evaluation metric. It measures the 1-Wasserstein distance between the empirical distributions of energy values computed from generated and ground truth samples. This metric is universally applicable across all sampling tasks and effectively captures discrepancies in the energy distributions regardless of the underlying state space and Markov processes involved.

**Magnetization 1-Wasserstein ( $M\text{-}\mathcal{W}_1$ ).** For the discrete Ising model, we additionally employ the magnetization 1-Wasserstein distance. Magnetization for a given spin configuration  $x \in \{-1, 1\}^{L \times L}$  is defined as the average spin:

$$M(x) = \frac{1}{L^2} \sum_i x_i. \quad (185)$$

This metric assesses the discrepancy in magnetization distributions between generated and ground truth samples. Particularly in low-temperature scenarios (e.g.,  $\beta = 0.4$ ), the system exhibits distinct modes around extreme magnetization values, making this metric especially sensitive to capturing difficulties in multimodal sampling.

Table 4: The best hyper-parameters combination for EGM and Bootstrapping (BS). Flow LR stands for the learning rate for the learned intermediate estimator.

Tasks	Method	Hidden dim.	# of layers	Flow LR	$\epsilon$
Ising $5 \times 5$ , $\beta = 0.2$	EGM	256	3	-	-
	BS	256	3	$2 \times 10^{-3}$	0.05
Ising $5 \times 5$ , $\beta = 0.4$	EGM	256	3	-	-
	BS	256	3	$10^{-3}$	0.05
Ising $10 \times 10$ , $\beta = 0.2$	EGM	1024	3	-	-
	BS	512	3	$10^{-3}$	0.05
Ising $10 \times 10$ , $\beta = 0.4$	EGM	256	3	-	-
	BS	2048	3	$10^{-3}$	0.05
GB-RBM	EGM	128	6	-	-
	BS	128	6	$10^{-3}$	0.01
JointDW4	EGM	128	6	-	-
	BS	128	6	$10^{-3}$	0.01
JointMoG	EGM	128	6	-	-
	BS	128	6	$10^{-3}$	0.01

**Sample 2-Wasserstein ( $x-\mathcal{W}_2$ ).** Specifically used for the GB-RBM task, the sample 2-Wasserstein distance evaluates discrepancies between the empirical distributions of generated and ground truth samples projected onto the first two continuous dimensions. A high sample 2-Wasserstein distance coupled with low energy 1-Wasserstein may indicate mode collapse within certain low-energy modes. Due to interpretability concerns (e.g., a poor sampler generating trivial solutions might misleadingly score well), we do not employ this metric for tasks beyond GB-RBM.

### D.3 Experimental setup

We performed a grid search to determine the optimal hyperparameters for each experimental task and method, evaluating each configuration using three random seeds.

As a baseline, we report the performance of a traditional Gibbs sampler [13]. Specifically, we ran Gibbs sampling with four parallel chains, each performing 6000 steps, collecting a total of 24,000 samples. For evaluation purposes, we uniformly subsampled 2000 samples from this set.

Across experiments, we employed 2000 Monte Carlo samples for estimations and a training batch size of 300. Both EGM and bootstrapping utilized 100 outer-loop iterations, with each iteration collecting 2000 samples into a buffer with a maximum size of 10k. The inner-loop iterations were set to 100 for EGM and 1000 for bootstrapping. We adopted a linear masking schedule ( $\kappa_t = t$ ), a linear conditional OT schedule ( $\alpha_t = t$ ), and an exponential variance exploding (VE) schedule ( $\sigma_t = \sigma_{\max}(\frac{\sigma_{\min}}{\sigma_{\max}})^t$ ). All samplers were trained using the AdamW optimizer with an initial learning rate of  $10^{-3}$ , applying a cosine learning rate schedule with  $\eta_{\min} = 10^{-5}$ . Training was conducted on an NVIDIA-3090 GPU (24GB VRAM).

For bootstrapping, the intermediate energy model  $\mathcal{E}^\phi$  was trained with a separately tuned learning rate. Bootstrapping step sizes of  $\epsilon \in \{0.01, 0.05\}$  were evaluated, and an exponential moving average (EMA) was applied to stabilize estimates from  $\mathcal{E}^\phi$ .

In multi-modal tasks, we applied a weighted loss combining discrete transition rate matrix prediction errors and continuous drift prediction errors:  $L_{\text{EGM}} = \lambda_{\text{disc}} L_{\text{discrete}} + \lambda_{\text{conti}} L_{\text{continuous}}$ , with fixed coefficients  $\lambda_{\text{disc}} = 5.0$  and  $\lambda_{\text{conti}} = 1.0$ .

Additional task-specific details are provided below, and optimal hyperparameters are summarized in Table 4.

**Discrete Ising model.** We employed a 3-layer MLP with sinusoidal time embeddings for both the intermediate energy function and the transition rate matrix. Each discrete token representing spin values -1 or 1 was embedded in 4 dimensions. Following Gat et al. [41], the transition rate matrix  $u_t(y, x)$  was parametrized using a probability denoiser  $p_{1|t}(y|x)$  analogous to the  $x_1$ -prediction in the flow models. Hidden dimensions were explored within  $\{256, 512\}$ , with additional trials at  $\{1024, 2048\}$  for the  $10 \times 10$  Ising grid.



1008 **GB-RBM.** We utilized a 6-layer residual MLP with 128 hidden units, a 4-dimensional discrete  
1009 embedding, and a 64-dimensional continuous embedding. Discrete and continuous embeddings  
1010 were concatenated and fed into the shared 6-layer MLP. Separate predictor networks subsequently  
1011 estimated the continuous drift and discrete transition rate matrix. The conditional OT path performed  
1012 best for both EGM and bootstrapping. We clipped the regression target  $F_t$  at a maximum norm of 20  
1013 and the energy estimator  $\hat{\mathcal{E}}_t$  at 100 to stabilize training.

1014 **JointDW4.** The network architecture matched that used in GB-RBM. The conditional OT path again  
1015 yielded optimal performance for both methods. Regression targets  $F_t$  and energy estimates  $\hat{\mathcal{E}}_t$  were  
1016 clipped at maximum norms of 100 and 1000, respectively.

1017 **JointMoG.** We maintained the same 6-layer residual MLP structure as GB-RBM. The VE path  
1018 achieved superior performance for both methods, configured with  $\sigma_{\max} = 2.0$  and  $\sigma_{\min} = 0.01$ .  
1019 The regression targets and energy estimates were clipped to maximum norms of 100 and 1000,  
1020 respectively.

## 1021 E Limitations and Discussion

1022 We have presented an energy-driven training framework for continuous-time Markov processes  
1023 (CTMPs). Our method introduces an energy-based importance sampling estimator for the marginal  
1024 generator and proposes an additional bootstrapping scheme to reduce the variance of importance  
1025 weights. By lowering this variance, we demonstrate that the bootstrapping approach significantly  
1026 enhances the sampler’s performance.

1027 **Limitations of our work.** Despite the strengths of our approach, several limitations remain. First, we  
1028 have not extensively evaluated the method on high-dimensional tasks due to limited computational  
1029 resources. While our framework performs well on benchmarks of moderate scale, its scalability  
1030 to complex high-dimensional domains—such as protein conformer generation—remains an open  
1031 question.

1032 Second, we observe that the training process can be unstable. We hypothesize that this instability  
1033 stems from the simultaneous optimization of the CTMP and the energy model. This joint training  
1034 often leads to degraded sampling performance. We apply exponential moving average updates to the  
1035 energy model, which empirically stabilizes training. Nonetheless, further investigation is required to  
1036 improve robustness.

1037 Third, our estimator incurs bias due to self-normalized importance sampling and the potential  
1038 mismatch between the proposal and the true posterior distributions. This bias may compromise the  
1039 accuracy of generator estimation, particularly when the proposal diverges significantly from the  
1040 posterior. Although the bootstrapping scheme helps reduce this mismatch, its effectiveness depends  
1041 on the intermediate energy estimator’s quality, which may introduce additional bias.

1042 **Comparison to LEAPS.** We compare our method to LEAPS [17], a neural sampler designed for  
1043 discrete spaces. Our framework is more general in that it applies to arbitrary state spaces and Markov  
1044 processes, including both continuous and discrete cases, whereas LEAPS is limited to discrete  
1045 domains. Even when instantiated with a discrete sampler, EGM and LEAPS differ fundamentally.  
1046 EGM relies on the prescribed conditional probability paths that mix the target distribution, while  
1047 LEAPS is built on the escorted transport with a temperature annealing. In continuous domains,  
1048 it has been shown that geometric annealing paths can lead to optimal drifts with high Lipschitz  
1049 constants [42], which limits sampler performance; whether a similar issue arises in discrete spaces  
1050 remains an open question.

1051 Additionally, EGM does not utilize an MCMC kernel (analogous to Langevin preconditioning in  
1052 continuous settings), whereas LEAPS explicitly relies on this mechanism. We believe exploring both  
1053 directions—leveraging and omitting Langevin preconditioning or MCMC kernels—offers promising  
1054 avenues for future research.

## F Additional results

### F.1 Additional qualitative results

We provide additional qualitative results for the experiments in Section 4. In Figure 6, we plot the energy histogram of the GB-RBM and JointDW4 compared to the ground truth sample. The Gibbs sampler baseline, EGM, and Bootstrapping match the ground truth energy histogram. However, the Gibbs sampler on GB-RBM suffers from mode collapse as demonstrated in Figure 3.

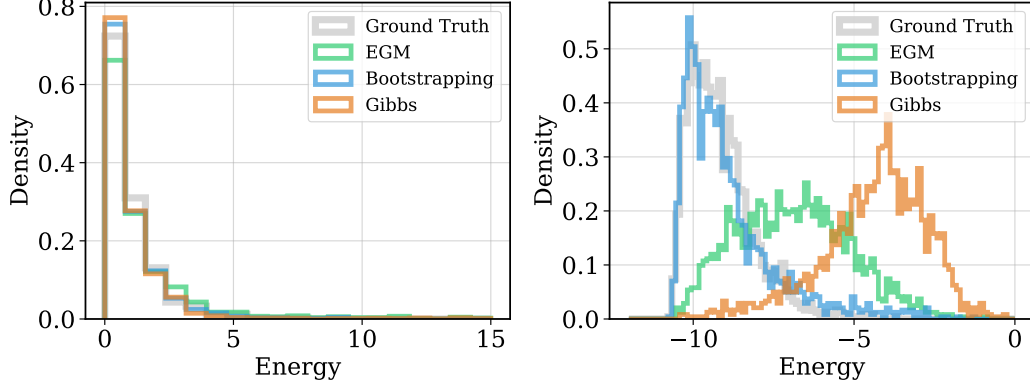


Figure 6: Energy histograms of various samplers on the GB-RBM (left) and JointDW4 (right).

### F.2 Effective sample size of our MC estimator

We present quantitative evidence of the bootstrapped estimator’s superiority over the naive EGM. Since the true marginal generator is intractable, we assess estimator quality via the effective sample size (ESS):

$$ESS = \frac{(\sum_{i=1}^n \tilde{w}_i)^2}{\sum_{i=1}^n \tilde{w}_i^2} \frac{1}{n} \quad (186)$$

where  $\tilde{w}_i$  denotes the unnormalized importance weight with the  $i$ -th proposed sample and  $n$  is the total number of MC samples. We report the *normalized* ESS to indicate the fraction of effectively used samples. Figure 7 shows the average normalized ESS over the course of training. The bootstrapped estimator maintains a significantly higher ESS during training, confirming its improved utilization of proposed samples compared to the naive EGM.

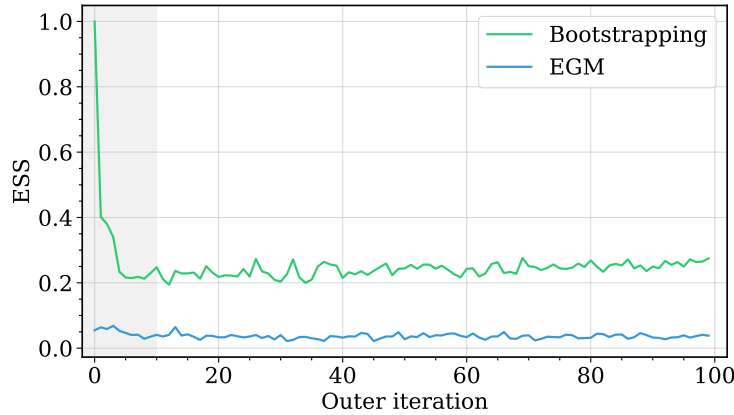


Figure 7: Effective sample size (ESS) of the Monte Carlo estimator during training on the Ising model ( $10 \times 10$ ,  $\beta = 0.4$ ). ESS is evaluated at each regression point and then averaged across all points. In the early training phase (shaded region), the energy model is insufficiently trained, so ESS estimates are unreliable.