COUNT-CONSERVING DIFFUSION FOR POINT PROCESSES IN TIME AND SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Planning often needs synthetic event scenarios that meet exact counts in several regions. Existing generative models control such counts at best on average, and correcting their samples afterwards either leaves overlapping counts wrong or meets them by rewriting the sample. We formalize this problem and develop a theory of count conservation for point-process diffusion. Viewing branching diffusion as a Wasserstein–Fisher–Rao gradient flow, we derive the projection of this flow that keeps every expected count, and show through the quadratic variation of the counts that its births and deaths still let each sample's counts drift. Conservation on every sample path requires every jump to lie in the null space of the count map. On a table of cell counts, the target factorizes into a table law and a position law shared by all budgets, and this condition admits three kinds of count-neutral updates and a Markov-basis chain that provably converges under overlap. Count-Conserving Diffusion is built from these updates alone: every sample meets all counts at every step, and one network serves every budget. On California wildfire scenarios with 33 overlapping coverage areas, global projection of DSTPP samples rewrites a median of 68–84% of the output events. Our samples meet every count without correction and are closer to the real data distribution.

1 INTRODUCTION

Planning with event data often requires scenarios with prescribed workloads rather than merely plausible samples from an unconditional point process: Hospitals test bed capacity against patient arrivals of a given volume (Harper & Shahani, 2002), ambulance services plan for emergency calls with specified workloads across service regions (Zhou et al., 2015), and wildfire agencies station aircraft for a given number of ignitions within reach of each base (Opitz et al., 2020), such as 30 near base A and 21 near base B (Figure 1). A scenario with the wrong workload is not a slightly imperfect sample but a different planning problem. A useful generator must meet every count in every sample and reproduce the structure that the counts leave open.

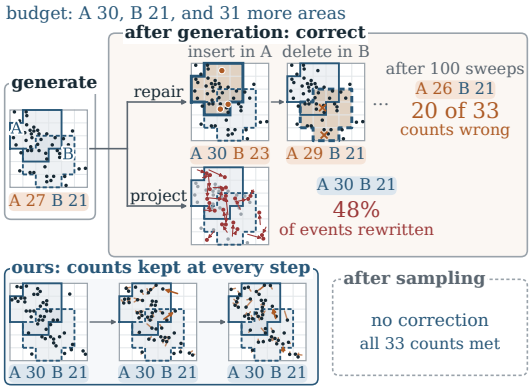


Figure 1: Correction breaks overlapping counts or rewrites the sample; conservation keeps all 33.

Diffusion models now generate whole event configurations: DSTPP (Yuan et al., 2023) models event times and locations jointly, ADD-THIN (Lüdke et al., 2023) and Point Set Diffusion (Lüdke et al., 2025) denoise entire point sets, and branching diffusion (Yang et al., 2026) moves events while births and deaths adapt their number. None takes prescribed counts as a hard input, and the births and deaths of branching diffusion change exactly the numbers that a budget fixes. *If the required counts are known in advance, why not simply enforce them throughout generation?*

For a single total, direct enforcement is straightforward; the difficulty lies in several overlapping counts. Exact conditioning of a trained chain would need, at every step, the probability that the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

finished sample meets the budget, which the chain does not provide (Section 3.3). In practice one generates and then corrects, but correction restores the counts and not the distribution, since its edits follow a rule rather than the data. Under overlap, one edit also changes several counts, since one ignition can lie within reach of up to six bases. In the wildfire week of Figure 1, repairing the 33 areas one by one leaves 20 counts wrong after 100 sweeps, and a minimum-cost projection meets all counts only by moving or replacing 48% of the events. We therefore need a generative process whose every update keeps all counts while it still learns the conditional distribution.

Our answer extends the theory behind branching diffusion. Yang et al. (2026) describe their process by its intensity, the expected density of events, which follows a Wasserstein–Fisher–Rao (WFR) gradient flow: mass is transported as events move and created or removed by births and deaths, along the fastest route to a reference. A budget constrains the mass of each region, and we show that, with transport inside cells, restricting the flow to directions that keep every regional mass gives the fastest such route, so every expected count stays fixed. A sample, however, follows the flow only on average: each birth or death in a region adds to the variance of its count, so balanced births and deaths keep the mean but let every sample drift. Counts hold in every sample only if no single step changes them, as for the conservation laws of chemical reaction networks. On small cells this condition becomes concrete: a step may move or redraw events within their cells, or shift events between cells in combinations that cancel in every region. Under overlap, a finite set of such combinations, a Markov basis from algebraic statistics, connects all admissible allocations. Given these allocations, positions follow one distribution that no budget changes.

We build a diffusion model from these steps alone. Noising moves and redraws events within their cells, and a network learns to reverse it given the cell counts. The cell counts come from a fitted count model: they are drawn exactly under partition budgets and, under overlap, by a Markov chain over admissible allocations whose Markov-basis steps make it converge. Every sample therefore meets all counts at every step, and one network serves every budget.

Our contributions are:

- We introduce event generation under exact, possibly overlapping count budgets and formulate it as sampling on the feasible slice of counting-measure space.
- We extend the WFR view of branching diffusion from expected to pathwise conservation of counts. The theory characterizes the count-preserving steps, factorizes the target, splits the sampling error exactly and yields a convergent chain over cell counts.
- We propose Count-Conserving Diffusion, which outperforms every exactly feasible baseline on five datasets under five constraint types, including overlapping coverage budgets.

2 Related Work

Generative models for event data. Neural point-process models first generated events one at a time, each conditioned on the history (Du et al., 2016; Shchur et al., 2020; Chen et al., 2021); DSTPP (Yuan et al., 2023) and later diffusion forecasters (Zhou et al., 2025; Zeng et al., 2024) keep this autoregressive structure while modeling each event's time and location with a diffusion model. A second line generates a whole window at once: ADD-THIN (Lüdke et al., 2023) and Point Set Diffusion (Lüdke et al., 2025) corrupt and restore complete point sets, EventFlow (Kerrigan et al., 2026) and EditTPP (Lüdke et al., 2026) transport them by flow matching, and Branching Diffusion (Yang et al., 2026) moves events and creates or removes them along a Wasserstein–Fisher–Rao gradient flow of intensities. In both lines, the number of events in a region is an outcome of sampling: autoregressive models fix it only once the window is complete, and set-level models control it at most through its expectation. We instead treat these numbers as constraints that every sampling step must respect.

Constrained and conserving generation. Constraints enter diffusion models either softly or by construction. Guidance (Dhariwal & Nichol, 2021; Ho & Salimans, 2022) and posterior sampling (Chung et al., 2023) steer a trained sampler toward a condition, and sequential Monte Carlo (Wu et al., 2023) approaches the exactly conditioned process as the number of particles grows, yet none guarantees the constraint in every sample. Hard constraints are instead built into the state space or the dynamics: reflected and mirror diffusion keep continuous states inside a constraint set (Lou & Ermon, 2023; Fishman et al., 2023; Liu et al., 2023), constrained discrete diffusion adds

an optimization step to reverse sampling (Cardei et al., 2025), and null-space methods for inverse problems keep linear measurements fixed by updating only what they leave free (Wang et al., 2023). Closest to our setting are samplers that conserve a single quantity: the total mass in discrete spatial diffusion (Santos et al., 2025), or the particle number in birth–death Langevin sampling (Lu et al., 2019), which pairs every deletion with a duplication. We carry this conservation principle to point sets of random size under many, possibly overlapping integer counts.

Conditioning on counts in statistics. Spatial and algebraic statistics have long sampled under fixed counts. Given its total, a Poisson process places its points independently (Kingman, 1993), and fixed-number Markov chain Monte Carlo explores point patterns with a given number of points (Geyer & Møller, 1994; Møller & Waagepetersen, 2004); aggregated-data Cox models go further and impute event locations from counts in areal units (Li et al., 2012; Diggle et al., 2013; Taylor et al., 2018). For integer tables with fixed linear statistics, Markov bases connect all tables that share these statistics (Diaconis & Sturmfels, 1998), and network tomography uses such chains to sample origin–destination traffic from overlapping link counts (Vardi, 1996; Tebaldi & West, 1998; Hazelton, 2015). These samplers are exact for a given parametric model but must be run again for every new set of counts. We adopt their table moves for the cell counts and learn the event positions within cells once, so that a single network serves every budget.

3 Preliminaries and Problem Formulation

3.1 Diffusion models for event data

An event is a point $z = (s, t)$ in $\mathcal{Z} = \mathcal{S} \times [0, T]$, a location in a bounded spatial domain $\mathcal{S} \subset \mathbb{R}^2$ and a time in a window of length T. A configuration of n events, such as a week of wildfire ignitions, is the counting measure $N = \sum_{i=1}^{n} \delta_{z_i}$, and $N(E)$ counts its events in a set $E \subseteq \mathcal{Z}$; the number n varies between windows. We treat training windows as independent draws from an unknown law P_{data} whose intensity λ gives the expected number of events in every set: $\mathbb{E}[N(E)] = \int_E \lambda(z)\,\mathrm{d}z$.

A diffusion model (Sohl-Dickstein et al., 2015; Ho et al., 2020) corrupts a configuration $N^0 \sim P_{\mathrm{data}}$ in K steps until N^K follows a reference law Q. Generation reverses the chain: it draws $N^K \sim Q$ and applies learned kernels $\kappa_k^\theta(N^k, \mathrm{d}N^{k-1})$, $k = K, \dots, 1$, with network parameters θ.

Because n varies, the corruption must also add and remove events. Branching diffusion (Yang et al., 2026) moves events and lets each one, independently of the others, give birth or die; its K steps discretize a diffusion time τ, and $Q = \mathrm{Pois}(\nu)$ is a Poisson process with a reference intensity ν. Starting from $\lambda_0 = \lambda$, the intensity λ_τ of the corrupted configurations follows the Wasserstein–Fisher–Rao (WFR) gradient flow of the generalized Kullback–Leibler divergence $\mathrm{GKL}(\lambda_\tau \| \nu) = \int (\lambda_\tau \log \frac{\lambda_\tau}{\nu} - \lambda_\tau + \nu)\,\mathrm{d}z$,

$$\partial_\tau \lambda_\tau = \underbrace{\nabla \cdot \Big(\lambda_\tau \nabla \log \frac{\lambda_\tau}{\nu} \Big)}_{\text{transport}} + \underbrace{\alpha_\tau\, \lambda_\tau}_{\text{growth}}, \qquad \alpha_\tau = -\log \frac{\lambda_\tau}{\nu}. \tag{1}$$

WFR writes a change of intensity as transport with a velocity v plus growth at a rate a, $\partial_\tau \lambda = -\nabla \cdot (\lambda v) + a\lambda$, and charges it $\int (\|v\|^2 + a^2)\,\lambda\,\mathrm{d}z$ (Chizat et al., 2018a; Liero et al., 2018); equation 1 is the steepest descent of GKL in this geometry, with $v = -\nabla \log(\lambda_\tau/\nu)$ and $a = \alpha_\tau$. At the sample level, each event follows Langevin dynamics toward ν and gives birth at rate $[\alpha_\tau]_+$ or dies at rate $[-\alpha_\tau]_+$ (Appendix A.1), and a network learns to reverse these moves, births and deaths. The flow therefore governs the intensity, and with it every expected count $\int_E \lambda_\tau\,\mathrm{d}z$, but not the count of any single sample.

3.2 Generation under count budgets

Let $R_1, \dots, R_M \subseteq \mathcal{Z}$ be regions, which may overlap, such as the reach of a firefighting base over one week. The count map $\Phi(N) = (N(R_1), \dots, N(R_M))$ records the workload of each region, and a *budget* $b \in \mathbb{Z}_+^M$ of nonnegative integers prescribes it: a week must contain 30 ignitions in area A and 21 in the overlapping area B (Figures 1 and 2a). For a budget with $P_{\mathrm{data}}(\Phi(N) = b) > 0$, the target is the data law conditioned on meeting it,

$$P_\star^b = P_{\mathrm{data}}\big(\,\cdot \mid \Phi(N) = b\big), \qquad \text{supported on the feasible slice } \mathcal{F}_b = \{N : \Phi(N) = b\}. \tag{2}$$

Every sample must lie in $\mathcal{F}_b$; how the samples spread over $\mathcal{F}_b$ must be learned from data.

A count depends on an event only through the regions that contain it, and cells make this explicit. Choose disjoint cells $C_1, \dots, C_J$, such as the boxes of a space–time grid, that cover $\mathcal{Z}$ and make every region a union of cells; a region that does not follow the grid, such as a base's reach, is replaced by the cells whose centers it contains, and from then on R_m, its budget and the target of equation 2 refer to these cells, on which counts are exact. The table of cell counts $x(N) = (N(C_1), \dots, N(C_J))$ then determines every count through $\Phi(N) = H\,x(N)$, with $H_{mj} = \mathbf{1}\{C_j \subseteq R_m\}$, and the budget admits the *feasible tables* $\mathcal{X}_b = \{x \in \mathbb{Z}_+^J : Hx = b\}$. A configuration is thus a table plus event positions within cells (Figure 2b), and feasible tables differ by integer null vectors of H. Only $\Phi(N) = H\,x(N)$ with an integer matrix H is used below, so everything extends to additive statistics $\sum_i \varphi(z_i)$ with integer φ constant on cells, such as weighted counts or marks.

The target splits in the same way. Writing $X = x(N)$ for the random table, the event $\{\Phi(N) = b\}$ is the disjoint union of the events $\{X = x\}$ over $x \in \mathcal{X}_b$, so

$$P_\star^b = \sum_{x \in \mathcal{X}_b} w_\star^b(x)\, P_{\mathrm{data}}(\cdot \mid X = x), \qquad w_\star^b(x) \propto P_{\mathrm{data}}(X = x)\, \mathbf{1}\{x \in \mathcal{X}_b\}, \tag{3}$$

summing over tables of positive probability. The budget only truncates one table law to $\mathcal{X}_b$; given the table, positions follow $P_{\mathrm{data}}(\cdot \mid X = x)$ under every budget whose regions are unions of the same cells. Section 4 exploits this factorization with one table law, truncated to $\mathcal{X}_b$ for each budget, and one position model shared by all budgets.

Partitions and overlap. In a *partition* budget the regions are disjoint and cover $\mathcal{Z}$, so each event enters exactly one count and $\mathcal{X}_b$ is a product of discrete simplices, one per region. On a space–time grid, a partition budget can fix the total count (Total), the counts per time slice (Temporal), the counts per spatial column summed over time (Spatial), or the whole table (Joint, $H = I$). Under *overlap*, as for the reach of neighboring bases (coverage budgets in Section 5.3), a cell can lie in several regions and an event enters several counts. $\mathcal{X}_b$ is then the set of integer points of a polyhedron without product structure, and a feasible table can no longer be drawn region by region.

3.3 Meeting a budget with existing generators

An existing generator can be steered toward a budget, conditioned on it, or corrected after sampling. Steering, by taking b as an input, by guidance (Dhariwal & Nichol, 2021; Ho & Salimans, 2022), posterior sampling (Chung et al., 2023) or count penalties, moves samples toward the counts but never enforces $\Phi(N) = b$. Exact conditioning is a Doob transform (Doob, 1957): it reweights each kernel κ_k^θ by $h_{k-1}(N^{k-1})/h_k(N^k)$, where $h_k(N) = \Pr_\theta(N^0 \in \mathcal{F}_b \mid N^k = N)$ is the probability that the trained chain ends in $\mathcal{F}_b$ from N, and tilts the start law Q by h_K, which yields the chain's own conditional law (Appendix A.2). A trained chain does not provide h_k, and estimating it takes many continuations or particles (Wu et al., 2023). The normalizer $\Pr_\theta(N^0 \in \mathcal{F}_b)$ of this transform is also the acceptance rate of rejection sampling, small when many counts are fixed: samples that DSTPP (Yuan et al., 2023) draws without the budget miss it in every wildfire test week (Section 5.3).

Correction edits a generated configuration until it meets the budget. *Repair* deletes random events from regions above their count and inserts uniform events into regions below it, either once after sampling or after every reverse step; under a partition budget it succeeds with $\|b - \Phi(N)\|_1$ edits. *Projection* returns the feasible configuration closest to the sample, moving, inserting or deleting events at minimal total transport cost. A correction Π of final samples that leaves feasible samples unchanged maps the law G of a generator to the law $\Pi_\# G$ of its corrected samples,

$$\Pi_\# G = G(\mathcal{F}_b)\, G(\cdot \mid \mathcal{F}_b) + \big(1 - G(\mathcal{F}_b)\big)\, \Pi_\# \big[G(\cdot \mid \mathcal{F}_b^c)\big], \tag{4}$$

so even a perfect generator, $G = P_{\mathrm{data}}$, yields $P_\star^b$ only if Π maps its infeasible samples onto $P_\star^b$ itself (Lemma 1); when $G(\mathcal{F}_b)$ is small, the correction rule shapes most of the output.

Overlap hampers correction, because one edit changes every count whose region contains the edited event: in Figure 1, inserting events into A to reach its 30 raises B to 23. We therefore ask for a sampler whose every update stays in $\mathcal{F}_b$, so that no correction is needed, and whose updates are learned from data, so that its final law approaches $P_\star^b$. Samplers from spatial statistics (Section 2) meet the first requirement but place events by a parametric model rather than a learned one, and

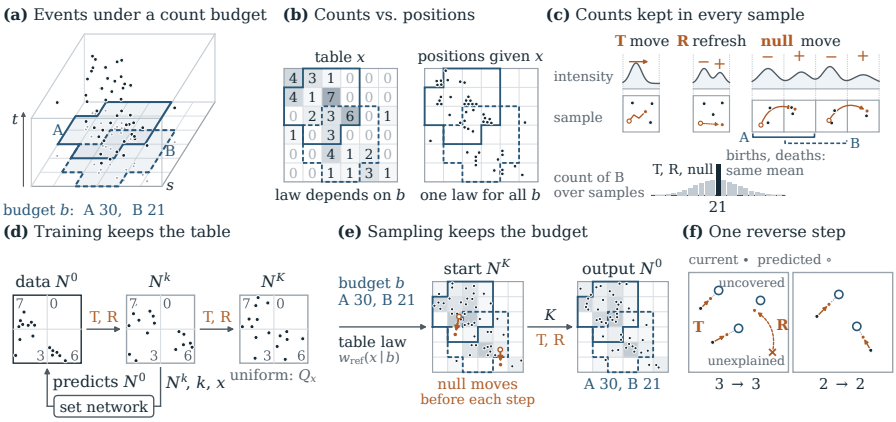


Figure 2: Count budgets and generation for areas A and B. (c) T, R and null moves keep every count in every sample; projected births and deaths keep only its mean. (f) A reverse step keeps cell counts.

branching diffusion fails it, since its births, deaths and moves across region boundaries change counts. Section 4 builds one by conditioning the reference of branching diffusion on the budget.

4 Count-Conserving Diffusion

The noisy end of a diffusion must itself meet the budget, so under a budget the reference of branching diffusion becomes $Q^b = Q(\cdot \mid \Phi(N) = b)$, and the divergence to descend becomes $\mathrm{KL}(P\|Q^b)$ over laws P on $\mathcal{F}_b$. Split by intensity (Section 4.1), this divergence tells which updates keep every count on every path; split by table (Section 4.2), which part a diffusion should descend. Section 4.3 builds the sampler (Figure 2).

4.1 Split by intensity: which updates keep a budget

At the level of intensities, a budget changes where branching diffusion may go, not the energy it descends. For a law P on $\mathcal{F}_b$ with intensity λ_P,

$$\mathrm{KL}\left(P \,\|\, Q^b\right) = \mathrm{KL}\left(P \,\|\, \mathrm{Pois}(\lambda_P)\right) + \mathrm{GKL}(\lambda_P \| \nu) + \log Q(\mathcal{F}_b), \tag{5}$$

where the intensity does not determine the first term and every λ_P has expected counts b (Appendix B). An intensity flow acts only on the second term, over intensities with these counts; with transport confined to cells, which keeps every region's mass under every budget on the grid, its WFR descent is

$$\partial_\tau \lambda_\tau = \nabla \cdot \left(\lambda_\tau \nabla \log \frac{\lambda_\tau}{\nu}\right) + \alpha_\tau^\perp \lambda_\tau, \qquad \alpha_\tau^\perp = \alpha_\tau - \textstyle\sum_m \beta_m \mathbf{1}_{R_m}, \tag{6}$$

where β makes $\alpha_\tau^\perp$ orthogonal to every indicator $\mathbf{1}_{R_m}$ in $\langle f, g\rangle_{\lambda_\tau} = \int f g \, \lambda_\tau \, \mathrm{d}z$; overlapping regions are coupled through $\int_{R_m \cap R_n} \lambda_\tau \, \mathrm{d}z$, so the projection is not computed region by region.

At the sample level, this growth would still be carried out by independent births and deaths. If update e of a jump process changes the configuration by ΔN_e at rate $r_e(N)$, with count increment $\Delta\Phi_e = \Phi(\Delta N_e)$, the counts $\Phi^\tau = \Phi(N^\tau)$ move over a short time $\Delta\tau$ with mean $\Delta\tau \sum_e r_e \Delta\Phi_e + o(\Delta\tau)$, the drift that projection removes on average, and mean square $\Delta\tau \sum_e r_e \|\Delta\Phi_e\|^2 + o(\Delta\tau)$, the growth of the counts' quadratic variation, which it does not remove.

Proposition 1 (Pathwise conservation)**.** *Let a pure-jump process with bounded rates and increments start in $\mathcal{F}_b$. (i) If it carries out $\alpha_\tau^\perp$ by independent births and deaths, the count of R_m gains quadratic variation at expected rate $\int_{R_m} |\alpha_\tau^\perp| \lambda_\tau \, \mathrm{d}z > 0$ unless $\alpha_\tau^\perp = 0$ a.e. on R_m. (ii) Its counts stay at b almost surely if and only if every possible update moves the table by a vector of* $\ker H \cap \mathbb{Z}^J$.

Part (i) is why births and deaths at the projected rates keep counts only on average: the count of B spreads around its mean (Figure 2c), and in a simulation only 5% of the samples keep both counts (Appendix B). Part (ii) is the sample-path counterpart of the projection: tangency to the budget

becomes an integer conservation law over jumps. A hard budget admits births and deaths only in combinations that change the table within $\ker H \cap \mathbb{Z}^J$, and each WFR component has such a form. **(T)** Transport confined to a cell moves an event within its cell $C(z)$. **(R)** Growth inside a cell, a death and a birth fused into one jump, *refreshes* an event: it redraws the event in its cell. A *null move* fuses growth across cells along a nonzero $\Delta x \in \ker H \cap \mathbb{Z}^J$: it deletes $[-\Delta x_j]_+$ events from each cell C_j and inserts $[\Delta x_j]_+$ there, provided $x + \Delta x \geq 0$. The null vectors of minimal support are the *circuits* of H, and these updates realize the flow:

Proposition 2 (Realization of the count-keeping flow)**.** *Formally, (i) the flow equation 6 keeps every expected count and, among directions with transport in cells that do, descends GKL most steeply in WFR; (ii) its cell growth $g_j = \int_{C_j} \alpha_\tau^\perp \lambda_\tau \, \mathrm{d}z$ has $Hg = 0$ and is a positive combination of circuits with the signs of g; (iii) pairing removal with creation within cells (R) and along these circuits (null moves) preserves counts b on every path and realizes the flow under the mean-field assumptions stated below.*

Part (iii) holds in the mean-field sense of branching diffusion, under a mass-action closure for circuits that remove several events (Appendix B), and pairs deaths with births along circuits, whereas birth–death Langevin sampling (Lu et al., 2019) pairs them with duplications to keep one total.

4.2 Split by table: a diffusion on positions

With w_P the table law of P and $P_x = P(\cdot \mid X = x)$, the chain rule over the table gives

$$\mathrm{KL}\left(P \,\|\, Q^b\right) = \mathrm{KL}\left(w_P \,\|\, w_Q^b\right) + \textstyle\sum_x w_P(x)\, \mathrm{KL}\left(P_x \,\|\, Q_x\right), \tag{7}$$

where w_Q^b, the table law of Q^b, restricts independent Poisson cell counts to $\mathcal{X}_b$, and $Q_x = Q(\cdot \mid X = x)$ places x_j independent events uniformly in each cell C_j when ν is constant on cells. Since Q_x does not depend on how ν distributes mass among cells, the table term can take its reference from data: mixing Poisson references with cell masses fitted to training tables gives the table law $w_{\mathrm{ref}}(\cdot \mid b)$ on $\mathcal{X}_b$ (Section 4.3). The position term's reference Q_x, and the law $P_{\mathrm{data}}(\cdot \mid X = x)$ that its reverse must reach, are by equation 3 the same for every budget on these cells. A diffusion that descends only this term therefore keeps the table, uses only (T) and (R) (at a fixed table $H = I$ and $\ker H = \{0\}$), and needs one network, trained without budgets, for every budget.

Forward chain. At a fixed table the regions are the cells, and the intensity flow equation 6 for the position term has two parts, realized by (T) and (R) without closure (Proposition 2). Its transport is a heat flow reflected at the cell walls when ν is constant there, whose exact transition folds a Gaussian step back into the cell. Its growth needs the unknown intensity; among kernels that delete and redraw an event within its cell without λ_τ, uniform refresh, $\partial_\tau \lambda_\tau = p_\tau(\bar{\lambda}_j - \lambda_\tau)$ toward the uniform intensity $\bar{\lambda}_j$ with the cell's mass, is the only one whose intensity flow equals this growth linearized at $\bar{\lambda}_j$, up to the speed p_τ (Appendix C), and it keeps Q_x invariant, so the position term never increases (Lemma 2). The parts commute, since heat flow keeps the uniform law and refresh forgets the position, so one step of both is a mixture (Figure 2d):

$$z_i^k = \begin{cases} \text{a uniform draw from } C(z_i^{k-1}) & \text{with probability } p_k, \\ \mathrm{Reflect}_{C(z_i^{k-1})}\left(z_i^{k-1} + \sigma_{\mathrm{f}} \odot \xi_i^k\right), \;\; \xi_i^k \sim \mathcal{N}(0, I) & \text{otherwise,} \end{cases} \tag{8}$$

where σ_{f} sets each coordinate's scale, $\mathrm{Reflect}_C$ folds coordinates into the box C, and p_k is the refresh probability of step k. With $p_K = 1$ the chain ends at Q_x (Lemma 2), and each event's posterior is explicit. The chain forgets the positions, which no budget on these cells constrains, and keeps the table, through which every such budget acts; branching diffusion forgets both.

Training. A set network (Zaheer et al., 2017) sees a noisy configuration, its table and the step, and predicts for every event z_i^k a displacement folded into its own cell, giving predicted events $\widehat{y}_i$ and $\widehat{N}_\theta^0 = \sum_i \delta_{\widehat{y}_i}$ with the same table. As in branching diffusion we compare predicted and clean sets by transport; with equally many events in every cell, the loss is a balanced entropic transport cost within cells (Cuturi, 2013) (Appendix C).

4.3 SAMPLING UNDER A BUDGET

The forward chain carries the target equation 3 to $\sum_x w_\star^b(x)\, Q_x$; the reverse chain replaces the unknown $w_\star^b$ by $w_{\mathrm{ref}}(\cdot \mid b)$, a Poisson mixture fitted to training tables and truncated to $\mathcal{X}_b$ as equation 3 prescribes, whose misfit is the table term of Proposition 3 (Figure 2e). It is drawn exactly at the start under partition budgets and approximated by the null-move chain below under overlap (Appendix D.1).

Reverse steps. The forward chain moves events independently, so its reversal given the clean positions is Bayes' rule for each event: given its clean position z^0, an event was either refreshed at step k, and z^{k-1} is a draw from the forward law at step $k-1$, or moved, and z^{k-1} lies on a Gaussian bridge between z^k and z^0 unless an earlier refresh cut its link to z^0 (Appendix D.2). A reverse step evaluates this posterior at the prediction, as x_0-predicting diffusion models do (Ho et al., 2020). As the loss compares sets, the ith prediction need not be the clean position of the ith event, and refreshed events have none; an entropic unbalanced coupling Γ between current and predicted events supplies the correspondence (Chizat et al., 2018b; Séjourné et al., 2019) (Figure 2f). **(T)** An event takes the bridge step toward its barycentric target $\bar{y}_i = \sum_{i'} \Gamma_{ii'} \widehat{y}_{i'} / \sum_{i'} \Gamma_{ii'}$, or the free step, folded into its cell. **(R)** It is refreshed with the posterior probability of the refresh branch at its partner, $\bar{y}_i$ for its matched mass and otherwise an uncovered predicted event of its cell; its log-odds, the *deletion logit*, grow with the distance to the partner and come from the posterior, whereas branching diffusion learns a growth field. The replacement is drawn around the partner, in the same cell.

Null moves under overlap. Under overlap, $w_{\mathrm{ref}}(\cdot \mid b)$ cannot be drawn directly, and by Proposition 1 only null moves change the table without changing the counts. From a feasible table found by an integer program, a Metropolis–Hastings chain of null moves keeps $w_{\mathrm{ref}}(\cdot \mid b)$ invariant (Hastings, 1970); relocation pairs whose count changes cancel are its simplest proposals but can leave feasible tables unreachable, so occasional moves from a Markov basis, which connects the feasible tables of every budget (Diaconis & Sturmfels, 1998), make it converge (Proposition 4, Appendix D.4). Proposals are guided by the first prediction, out of cells whose events it leaves unexplained into cells whose predicted events are uncovered; with fixed weights, guidance changes the chain's speed, not its target. The chain runs before each reverse step, mostly early, while positions are still near uniform, and places moved events uniformly in their new cells.

Every state lies in $\mathcal{F}_b$ (Algorithm 1): the sampler does not estimate the function h_k of Section 3.3 but uses only transitions that keep the budget. By the chain rule of equation 7, the error $\mathrm{KL}(P_\star^b \| P_\theta^b)$ of its output law P_θ^b splits into a table term, set by w_{ref} and the table chain, and a position term, the only one the network affects under partition budgets (Proposition 3).

5 EXPERIMENTS

5.1 SETUP

Data and budgets. SYN-1 is a multimodal inhomogeneous Poisson process, and SYN-2 adds modes that vary between windows; the real data are earthquakes in Japan (EARTHQUAKE; U.S. Geological Survey, 2017), daily emergency medical calls in Seattle (EMS; City of Seattle, 2026) and, in Section 5.3, weekly wildfire ignitions in California. Partition budgets, taken from each test window, fix the Total, Temporal, Spatial or Joint counts of Section 3.2 on $8 \times 8 \times 8$ space–time cells.

Baselines and metrics. DSTPP (Yuan et al., 2023) and branching diffusion (Yang et al., 2026) are repaired after generation (DSTPP + Repair, BD + Repair); branching diffusion is also repaired after every reverse step (BD + Step Repair) or penalized toward the counts (BD + Count Penalty), and Uniform-Budget places events uniformly in the budget regions. MMD compares whole-window summaries (Gretton et al., 2012) and CMMD does so within budget cells; STSD matches generated and real windows one to one by unbalanced Sinkhorn divergences (Séjourné et al., 2019). Lower is better; all three are averaged over five training runs (Appendix E.1).

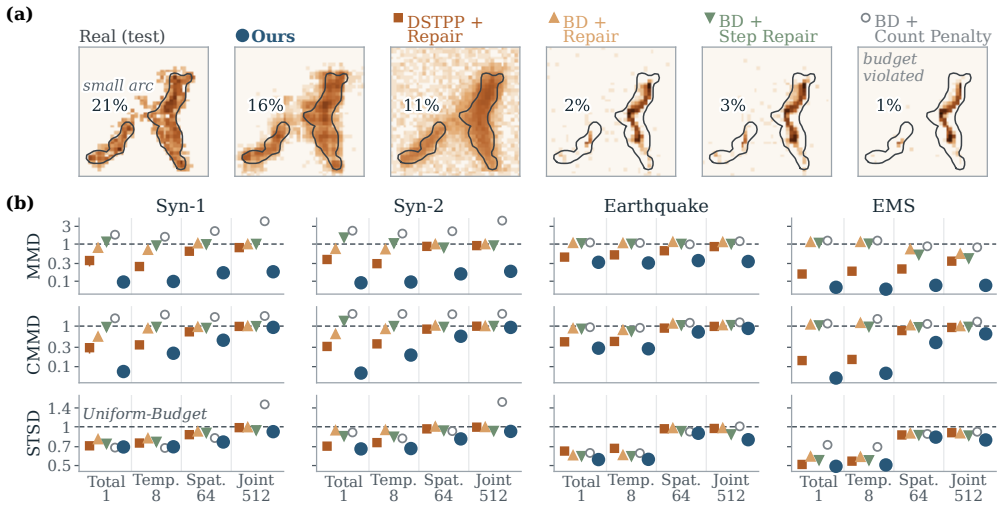


Figure 3: Partition budgets. (a) EARTHQUAKE, Total budget; contours hold 90% of real events; %: share in the small arc. (b) Ratios to Uniform-Budget (dashed), ± 1 s.d.; hollow: budget violated.

5.2 PARTITION BUDGETS

Figure 3b compares the methods on four datasets; within each panel, the budgets from left to right fix 1, 8, 64 and 512 counts. BD + Count Penalty (hollow markers) misses the budget in most of its samples, because a penalty on expected counts leaves the fluctuations of Proposition 1(i) in place. Among exactly feasible methods, ours is the most accurate throughout (Appendix E.2), and two patterns in the figure explain why.

Correction becomes more intrusive as budgets become finer. Repair needs $\|b - \Phi(N)\|_1$ edits, a number that never decreases when a partition is refined, so a growing part of each repaired sample is placed by the repair rule rather than by the generator, as equation 4 predicts. Accordingly, the MMD ratio of DSTPP + Repair (squares in the top row) roughly doubles from Total to Joint budgets on every dataset. Repairing after every reverse step (BD + Step Repair) does not help, because the same rule then edits every intermediate state.

The gain lies in how events are placed within cells. Under Joint budgets every exactly feasible method outputs the same table, so the table term of equation 7 is common to all methods and only the position term differs; our MMD nevertheless stays well below that of the baselines. Figure 3a shows samples on EARTHQUAKE under a Total budget: ours follow both seismic arcs, DSTPP + Repair spreads events beyond them, and the branching-diffusion baselines nearly empty the small arc (2–3% of the events, against 21% in the data).

5.3 WILDFIRE RESPONSE: OVERLAPPING COVERAGE BUDGETS

Planners station air tankers and helicopters at bases and need weekly ignition scenarios with a set workload in each base's reach (Figure 4a). On California ignitions from 2001 to 2024 (Short, 2026), with 156 test weeks, the budget fixes daily totals and weekly counts in the gridded areas of 16 air-attack and 17 helitack bases (California Department of Forestry and Fire Protection, 2026); a harder version fixes daily area counts (Appendix E.3).

Overlap changes how correction fails. We correct DSTPP, the strongest repaired baseline, in two ways (Figure 4b). An edit in one cell changes the count of every area that contains it, so local repair, which fixes the areas one at a time, keeps disturbing areas it has already fixed; in the week shown, 20 of the 33 counts are still wrong after 100 sweeps. Global projection avoids this by meeting all counts at once, but it must move or replace a share of the events that grows with the number of fixed counts, from a median of 29% under daily totals to 68% and 84% with weekly and daily area counts. Since DSTPP samples miss the coverage budgets in every week, only the second term of equation 4 remains, and the projection rule shapes every output.

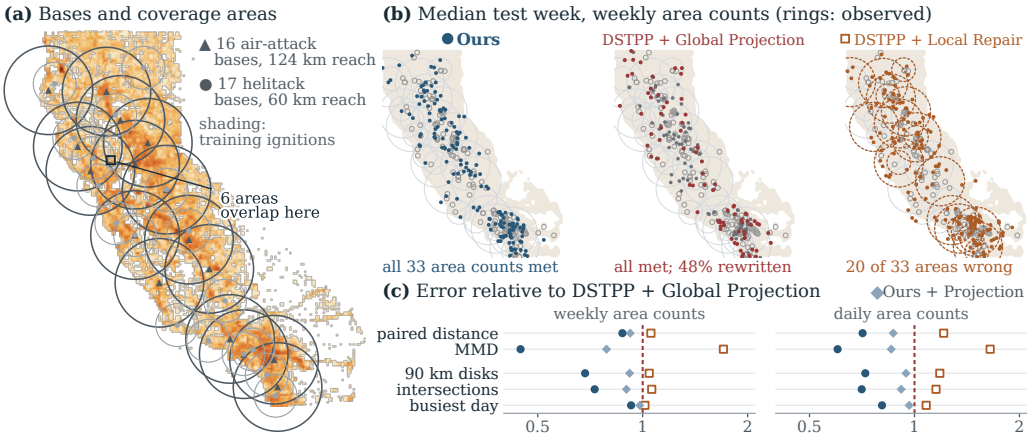


Figure 4: Coverage budgets. (b) Median-size week; red: rewritten events; dashed: areas left wrong. (c) Errors relative to projected DSTPP (dotted); diamonds: ours under daily totals, then projected.

The budget leaves open many quantities that planners need, such as the number of ignitions within 90 km of a base, in the region reached by two bases, or on the busiest day of a base. These disk queries cut across cells, so they depend on placement within cells, which projection decides by transport cost rather than by the data. Our samples answer them more accurately under both coverage budgets, with the smallest gains on the busiest day (Figure 4c and Appendix E.3).

The diamonds in Figure 4c keep our network but change the sampling route: they sample under the daily totals only, which fix no area, and project onto the coverage budget afterwards. All errors rise, so the benefit does not come from the network alone; conserving the counts during sampling lets the learned reverse steps place every event, whereas projection moves events by transport cost. The samples also do not collapse onto the actual week: the distance between two samples of a week is close to their distance to the actual week (ratios 0.97 and 0.90, where exact draws give one).

5.4 Sampling cost

Conserving counts costs more than repair under coarse budgets and less under the finest ones. Our reverse steps couple current and predicted events, so their cost grows with the number of events, whereas the cost of repair grows with the number of cells it must fix. Under Total budgets our sampler is about twice as slow as BD + Repair on windows of up to 100 events and up to nine times as slow on the largest EMS windows, but under Joint budgets, where repair must fix 512 cells, it is faster (Appendix E.4). A wildfire week takes about three seconds on one CPU, mostly for couplings and null moves.

6 Conclusion

We presented Count-Conserving Diffusion, a diffusion model for event data in which every update keeps prescribed, possibly overlapping counts, so that no correction is needed after sampling. Our analysis distinguishes conservation in expectation from conservation on every sample path: projecting the WFR flow of branching diffusion keeps expected counts, whereas exact counts require every jump to lie in the null space of the count map. On a table of cell counts, this condition turns a linear count constraint into a lattice condition on the table and leaves the event positions to one network shared by all budgets. In our experiments, correction degrades as a budget fixes more counts and as its regions overlap, while our samples meet every count and are closer to real data; on 33 overlapping wildfire areas, the advantage remains when the network is fixed and only the sampling route changes.

Our construction represents every region by cells of a grid, so regions that do not follow the grid are approximated at its resolution. Extending exact conservation to regions of arbitrary shape and to budgets given as ranges rather than exact counts is a natural next step.

AI USE STATEMENT

We used generative AI tools as assistants in this work. They helped us organize and summarize related literature, and helped write plotting code for the figures and format result tables. They also helped locate candidate public datasets. For the writing, AI tools helped polish the text and give feedback on clarity and consistency. We reviewed all AI-assisted content and take full responsibility for this paper.

REPRODUCIBILITY STATEMENT

Section 4 and Appendices B–D specify the model, the training objective and both samplers, and Appendix E lists data sources, splits, budgets, metrics and run counts. All propositions and lemmas are proved in Appendices A–D.

ETHICS STATEMENT

The wildfire and emergency-call records are public releases of government agencies (Short, 2026; City of Seattle, 2026); we use only event times and coordinates, aggregated into windows, and generate synthetic scenarios for planning rather than predictions about individuals.

REFERENCES

David F. Anderson and Thomas G. Kurtz. *Stochastic Analysis of Biochemical Systems*, volume 1.2 of *Mathematical Biosciences Institute Lecture Series*. Springer, 2015. doi: 10.1007/978-3-319-16895-1.

Satoshi Aoki, Hisayuki Hara, and Akimichi Takemura. *Markov Bases in Algebraic Statistics*. Springer Series in Statistics. Springer, New York, 2012.

California Department of Forestry and Fire Protection. CAL FIRE Facilities for Wildland Fire Protection – Public View. ArcGIS Online item 9e99328b43bc4e448ad5758790d17d4b, 2026. Accessed September 21, 2026.

Michael Cardei, Jacob K Christopher, Bhavya Kailkhura, Tom Hartvigsen, and Ferdinando Fioretto. Constrained discrete diffusion. *Advances in Neural Information Processing Systems*, 38, 2025.

Ricky T. Q. Chen, Brandon Amos, and Maximilian Nickel. Neural spatio-temporal point processes. In *International Conference on Learning Representations*, 2021.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. An interpolating distance between optimal transport and Fisher–Rao metrics. *Foundations of Computational Mathematics*, 18(1):1–44, 2018a.

Lénaïc Chizat, Gabriel Peyré, Bernhard Schmitzer, and François-Xavier Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018b.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

City of Seattle. Seattle Real Time Fire 911 Calls. Seattle Open Data portal, dataset kzjm-xkqj, 2026. Accessed September 19, 2026.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006. doi: 10.1002/047174882X.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, 1975.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021.

Persi Diaconis and Bernd Sturmfels. Algebraic algorithms for sampling from conditional distributions. *The Annals of Statistics*, 26(1):363–397, 1998.

Peter J. Diggle, Paula Moraga, Barry Rowlingson, and Benjamin M. Taylor. Spatial and spatio-temporal log-Gaussian Cox processes: Extending the geostatistical paradigm. *Statistical Science*, 28(4):542–563, 2013.

Joseph L. Doob. Conditional Brownian motion and the boundary limits of harmonic functions. *Bulletin de la Société Mathématique de France*, 85:431–458, 1957.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1555–1564, 2016.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and MMD using Sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 2681–2690, 2019. URL `https://proceedings.mlr.press/v89/feydy19a.html`.

Nic Fishman, Leo Klarner, Valentin De Bortoli, Emile Mathieu, and Michael Hutchinson. Diffusion models for constrained domains. *Transactions on Machine Learning Research*, 2023.

Charles J. Geyer and Jesper Møller. Simulation procedures and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics*, 21(4):359–373, 1994.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(1):723–773, 2012.

Paul R. Harper and Arjan K. Shahani. Modelling for the planning and management of bed capacities in hospitals. *Journal of the Operational Research Society*, 53(1):11–18, 2002.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. doi: 10.1093/biomet/57.1.97.

Martin L. Hazelton. Network tomography for integer-valued traffic. *The Annals of Applied Statistics*, 9(1):474–506, 2015.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Gavin Kerrigan, Kai Nelson, and Padhraic Smyth. EventFlow: Forecasting temporal point processes with flow matching. In *International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, 2026.

J. F. C. Kingman. *Poisson Processes*, volume 3 of *Oxford Studies in Probability*. Oxford University Press, 1993.

David A. Levin and Yuval Peres. *Markov Chains and Mixing Times*. American Mathematical Society, 2nd edition, 2017. doi: 10.1090/mbk/107.

Ye Li, Patrick Brown, Dionne C. Gesink, and Håvard Rue. Log Gaussian Cox processes and spatially aggregated disease incidence data. *Statistical Methods in Medical Research*, 21(5):479–507, 2012.

Matthias Liero, Alexander Mielke, and Giuseppe Savaré. Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures. *Inventiones Mathematicae*, 211 (3):969–1117, 2018.

Guan-Horng Liu, Tianrong Chen, Evangelos A. Theodorou, and Molei Tao. Mirror diffusion models for constrained and watermarked generation. In *Advances in Neural Information Processing Systems*, 2023.

Aaron Lou and Stefano Ermon. Reflected diffusion models. In *International Conference on Machine Learning*, pp. 22675–22701. PMLR, 2023.

Yulong Lu, Jianfeng Lu, and James Nolen. Accelerating Langevin sampling with birth-death. *arXiv preprint arXiv:1905.09863*, 2019.

David Lüdke, Marin Bilos, Oleksandr Shchur, Marten Lienen, and Stephan Günnemann. Add and thin: Diffusion for temporal point processes. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/b1d9c7e7bd265d81aae8d74a7a6bd7f1-Abstract-Conference.html.

David Lüdke, Enric Rabasseda Raventós, Marcel Kollovieh, and Stephan Günnemann. Unlocking point processes through point set diffusion. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4anfpHj0wf.

David Lüdke, Marten Lienen, Marcel Kollovieh, and Stephan Günnemann. Edit-based flow matching for temporal point processes. In *International Conference on Learning Representations*, 2026.

Jesper Møller and Rasmus Plenge Waagepetersen. *Statistical Inference and Simulation for Spatial Point Processes*. Chapman & Hall/CRC, Boca Raton, FL, 2004.

Thomas Opitz, Florent Bonneu, and Edith Gabriel. Point-process based Bayesian modeling of space–time structures of forest fire occurrences in Mediterranean France. *Spatial Statistics*, 40: 100429, 2020.

R. Tyrrell Rockafellar. The elementary vectors of a subspace of R^N. In *Combinatorial Mathematics and its Applications*, pp. 104–127. University of North Carolina Press, 1969.

Javier E. Santos, Agnese Marcato, Roman Colman, Nicholas Lubbers, and Yen Ting Lin. Discrete spatial diffusion: Intensity-preserving diffusion modeling. *Advances in Neural Information Processing Systems*, 38, 2025.

Thibault Séjourné, Jean Feydy, François-Xavier Vialard, Alain Trouvé, and Gabriel Peyré. Sinkhorn divergences for unbalanced optimal transport. *arXiv preprint arXiv:1910.12958*, 2019.

Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. In *International Conference on Learning Representations*, 2020.

Karen C. Short. Spatial wildfire occurrence data for the United States, 1992–2024 [FPA_FOD_20260615], 7th edition. Forest Service Research Data Archive, Fort Collins, CO, 2026.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265, 2015.

Benjamin M. Taylor, Ricardo Andrade-Pacheco, and Hugh J. W. Sturrock. Continuous inference for aggregated point process data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 181(4):1125–1150, 2018.

Claudia Tebaldi and Mike West. Bayesian inference on network traffic using link count data. *Journal of the American Statistical Association*, 93(442):557–573, 1998.

Luke Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4): 1701–1728, 1994. doi: 10.1214/aos/1176325750.

U.S. Geological Survey. Advanced National Seismic System (ANSS) Comprehensive Catalog. Earthquake Hazards Program, 2017.

Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American Statistical Association*, 91(433):365–377, 1996.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *International Conference on Learning Representations*, 2023.

Luhuan Wu, Brian L. Trippe, Christian A. Naesseth, David M. Blei, and John P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Chao Yang, Wenjie Shen, and Shuang Li. Branching diffusion for point processes in time and space. In *International Conference on Machine Learning*, 2026.

Yuan Yuan, Jingtao Ding, Chenyang Shao, Depeng Jin, and Yong Li. Spatio-temporal diffusion point processes. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3173–3184. ACM, 2023. doi: 10.1145/3580305.3599511.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in Neural Information Processing Systems*, 30, 2017.

Mai Zeng, Florence Regol, and Mark Coates. Interacting diffusion processes for event sequence forecasting. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 2024.

Wang-Tao Zhou, Zhao Kang, Ling Tian, Jinchuan Zhang, and Yumeng Liu. Non-autoregressive diffusion-based temporal point processes for continuous-time long-term event prediction. *Expert Systems with Applications*, 267:126210, 2025.

Zhengyi Zhou, David S Matteson, Dawn B Woodard, Shane G Henderson, and Athanasios C Micheas. A spatio-temporal point process model for ambulance demand. *Journal of the American Statistical Association*, 110(509):6–15, 2015.

APPENDIX

Each appendix supplements the main-text section named in its title. Propositions 1 and 2 are proved in Appendix B; Propositions 3 and 4, which concern the sampler, are stated and proved in Appendix D.

A PRELIMINARIES: WFR FLOW, CONDITIONING AND CORRECTION (SECTION 3)

A.1 THE WFR FLOW AND THE POISSON KL DIVERGENCE

The WFR gradient flow. The first variation of $\mathrm{GKL}(\lambda\|\nu)$ is $\phi = \log(\lambda/\nu)$. In the WFR metric, which charges a velocity v and a growth rate a by $\int(\|v\|^2 + a^2)\,\lambda\,\mathrm{d}z$, the steepest descent direction is $v = -\nabla\phi$ and $a = -\phi$ (Chizat et al., 2018a; Liero et al., 2018); the continuity equation $\partial_\tau\lambda = -\nabla\cdot(\lambda v) + a\lambda$ then gives equation 1, as derived by Yang et al. (2026). At the sample level, their chain moves events and lets each one independently die at rate $[-\alpha_\tau]_+$ or give birth at rate $[\alpha_\tau]_+$, and generation starts from a Poisson process with intensity ν.

Poisson KL and its limitation. For Poisson processes with intensities λ and ν on $\mathcal{Z}$, the density of the first with respect to the second at a configuration $N = \sum_i \delta_{z_i}$ is $\exp\{\int(\nu - \lambda)\,\mathrm{d}z\}\prod_i \lambda(z_i)/\nu(z_i)$. By Campbell's formula, $\mathbb{E}\sum_i f(z_i) = \int f\lambda\,\mathrm{d}z$ for any point process with intensity λ, so the expected logarithm of this density is $\mathrm{GKL}(\lambda\|\nu) = \int(\lambda\log\frac{\lambda}{\nu} - \lambda + \nu)\,\mathrm{d}z$ under the first process, the generalized KL divergence of Section 3.1. The same formula holds under any law P with intensity λ, which gives, whenever the terms are finite,

$$\mathrm{KL}\left(P \,\|\, \mathrm{Pois}(\nu)\right) = \mathrm{KL}\left(P \,\|\, \mathrm{Pois}(\lambda)\right) + \mathrm{GKL}(\lambda\|\nu).$$

The intensity determines the second term; its evolution alone does not determine the first, which vanishes only when P is Poisson: a process with exactly one uniform event on a unit domain and a Poisson process with mean one share their intensity, but not their law.

A.2 EXACT CONDITIONING AND CORRECTION

Exact conditioning. For the reverse chain of Section 3.1, with kernels κ_k^θ and start $N^K \sim Q$, let $h_k(N) = \Pr_\theta(N^0 \in \mathcal{F}_b \mid N^k = N)$ be the probability of ending feasible from state N, and $Z_b = \Pr_\theta(N^0 \in \mathcal{F}_b) > 0$. The Doob transform (Doob, 1957)

$$\kappa_k^b(N, \mathrm{d}N') = \kappa_k^\theta(N, \mathrm{d}N')\,\frac{h_{k-1}(N')}{h_k(N)}, \qquad Q^h(\mathrm{d}N) = \frac{h_K(N)\,Q(\mathrm{d}N)}{Z_b} \tag{9}$$

gives the law of the chain conditioned on $N^0 \in \mathcal{F}_b$; its normalizer Z_b is the acceptance probability of terminal rejection sampling.

Correction. A correction may be random, as when repair inserts uniform events, so we state the identity of Section 3.3 for a Markov kernel Π on configurations; a measurable map is the special case $\Pi(N,\cdot) = \delta_{\Pi(N)}$, and $\Pi_\#\mu$ in the main text denotes $\mu\Pi = \int\mu(\mathrm{d}N)\,\Pi(N,\cdot)$. Total variation is $\mathrm{TV}(\mu,\nu) = \sup_E|\mu(E) - \nu(E)|$.

Lemma 1 (Law after correction)**.** *Let G be a law on configurations with $0 < G(\mathcal{F}_b) < 1$, and let Π be a Markov kernel with $\Pi(N,\mathcal{F}_b) = 1$ for every N and $\Pi(N,\cdot) = \delta_N$ for $N \in \mathcal{F}_b$. Then*

$$G\Pi = G(\mathcal{F}_b)\,G(\cdot \mid \mathcal{F}_b) + \big(1 - G(\mathcal{F}_b)\big)\,G(\cdot \mid \mathcal{F}_b^c)\Pi.$$

If moreover $G(\cdot \mid \mathcal{F}_b) = P_\star^b$, as for $G = P_{\mathrm{data}}$, then $\mathrm{TV}(G\Pi, P_\star^b) = \big(1 - G(\mathcal{F}_b)\big)\,\mathrm{TV}\big(G(\cdot \mid \mathcal{F}_b^c)\Pi, P_\star^b\big)$, so $G\Pi = P_\star^b$ if and only if $G(\cdot \mid \mathcal{F}_b^c)\Pi = P_\star^b$.

Proof. For a measurable set E of configurations, $G\Pi(E) = \int_{\mathcal{F}_b} G(\mathrm{d}N)\,\mathbf{1}_E(N) + \int_{\mathcal{F}_b^c} G(\mathrm{d}N)\,\Pi(N,E) = G(\mathcal{F}_b)\,G(E \mid \mathcal{F}_b) + G(\mathcal{F}_b^c)\,[G(\cdot \mid \mathcal{F}_b^c)\Pi](E)$. With $q = G(\mathcal{F}_b)$ and $R = G(\cdot \mid \mathcal{F}_b^c)\Pi$, the assumption $G(\cdot \mid \mathcal{F}_b) = P_\star^b$ gives $G\Pi - P_\star^b = qP_\star^b + (1-q)R - P_\star^b =$

$(1-q)(R-P^b_\star)$, so $|G\Pi(E)-P^b_\star(E)| = (1-q)\,|R(E)-P^b_\star(E)|$ for every E; taking the supremum over E gives the TV identity, and $1-q>0$ gives the last claim. □

For repair, count each deletion and each insertion as one edit. Under a partition budget each edit changes exactly one count by one, so at least $\|b-\Phi(N)\|_1$ edits are necessary, and deleting surplus and inserting deficit events region by region attains this bound. Overlap removes this separability. For four cells, in A only, in both areas, in B only and in neither, with A, B and the total as counted regions (Figure 2c),

$$H=\begin{pmatrix}1&1&0&0\\0&1&1&0\\1&1&1&1\end{pmatrix},\qquad b=(1,1,2)^\top,\qquad \Delta x=(-1,1,-1,1)^\top.$$

The two relocations in Δx have nonzero count increments, but their sum is zero. A region-by-region repair and a global feasible projection therefore solve different problems, and in the former an edit for one region can break a count that an earlier edit fixed.

B Constrained WFR Geometry and Pathwise Conservation (Section 4.1)

The energy of a budget. Let $Q=\mathrm{Pois}(\nu)$ and $Q^b=Q(\cdot\mid\mathcal{F}_b)$, and let P be a law on $\mathcal{F}_b$ with intensity λ_P and $P\ll Q$. On $\mathcal{F}_b$, $\mathrm{d}Q^b/\mathrm{d}Q=1/Q(\mathcal{F}_b)$, so $\mathrm{KL}(P\|Q^b)=\mathrm{KL}(P\|Q)+\log Q(\mathcal{F}_b)$, and the identity of Appendix A.1 gives equation 5. Every such P has $\int_{R_m}\lambda_P\,\mathrm{d}z=\mathbb{E}_P[N(R_m)]=b_m$. Among intensities with these expected counts, $\mathrm{GKL}(\cdot\|\nu)$ is strictly convex and is minimized by the I-projection $\lambda_\infty=\nu\exp(\sum_m\eta_m\mathbf{1}_{R_m})$, with η set by the counts (Csiszár, 1975). It is also the rest point of the flow of Proposition 2: a rest point has $\nabla\log(\lambda/\nu)=0$ in every cell and $\alpha^\perp=0$, so $\log(\lambda/\nu)$ lies in the span of the region indicators. Relative to Q, $\mathrm{Pois}(\lambda_\infty)$ has density proportional to $\exp\{\sum_i\sum_m\eta_m\mathbf{1}_{R_m}(z_i)\}=\exp\{\eta^\top\Phi(N)\}$, which is constant on $\mathcal{F}_b$; hence $\mathrm{Pois}(\lambda_\infty)(\cdot\mid\mathcal{F}_b)=Q^b$. This identity holds for every η, so it does not make λ_∞ the intensity of Q^b. Under a partition budget it is: Q^b places b_m independent events with density proportional to ν in R_m, whose intensity $b_m\nu/\nu(R_m)$ on R_m is of the form νe^{η_m} with the right counts; a fixed table is the partition into cells. Under overlap it need not be. With three cells, in A only, in both and in B only, $\nu(C_j)=1$ and $b=(1,1)$, the feasible tables are $(1,0,1)$ and $(0,1,0)$, each with probability $\frac12$ under Q^b, so the intensity of Q^b has cell masses $(\frac12,\frac12,\frac12)$, whereas λ_∞ has cell masses (u,u^2,u) with $u+u^2=1$, that is, $(0.618,0.382,0.618)$. The two laws $\mathrm{Pois}(\lambda_\infty)$ and Q^b then differ in the first term of equation 5, which no intensity flow can change; the sampler therefore reaches the table law by a chain on tables (Section 4.3).

Keeping expected counts. If transport stays inside cells, a region's expected count changes only through growth, at rate $\int_{R_m}\alpha_\tau\lambda_\tau\,\mathrm{d}z=\langle\alpha_\tau,\mathbf{1}_{R_m}\rangle_{\lambda_\tau}$, with $\langle f,f'\rangle_\lambda=\int ff'\lambda\,\mathrm{d}z$; removing from α_τ its component along the region indicators stops these changes. For disjoint regions this centers α_τ within each region, and for the total count alone it centers α_τ globally, which gives the mass-conserving birth–death flow of Lu et al. (2019). We assume smooth positive λ_τ and ν with finite dissipation, and reflecting cell walls; the statements are formal in the sense of equation 1, since $\alpha^\perp_\tau$ depends on λ_τ.

Proof of Proposition 1. For a pure-jump process with bounded total rate and bounded count increments, update e occurs in an interval of length $\Delta\tau$ with probability $r_e(N)\Delta\tau+o(\Delta\tau)$, and two or more jumps contribute $o(\Delta\tau)$ to the first two moments. Therefore

$$\mathbb{E}[\Delta\Phi\mid N]=\Delta\tau\textstyle\sum_e r_e(N)\,\Delta\Phi_e+o(\Delta\tau),$$
$$\mathbb{E}[\Delta\Phi\,\Delta\Phi^\top\mid N]=\Delta\tau\textstyle\sum_e r_e(N)\,\Delta\Phi_e\Delta\Phi_e^\top+o(\Delta\tau);$$

the first is the drift and the trace of the second the mean square of Section 4.1. These are the drift of the counts and the rate of their quadratic variation, as for linear conservation laws of reaction networks (Anderson & Kurtz, 2015). For part (ii), the second moment is a sum of nonnegative terms, so it vanishes at N exactly when every update with $r_e(N)>0$ has $\Delta\Phi_e=0$. If an update with $\Delta\Phi_e\neq0$ has $r_e(N)>0$ at a state N reachable from $\mathcal{F}_b$, the process reaches N and makes

this update next with positive probability, and the counts change; if no such update exists, no jump changes them. Since $\Phi(N) = Hx(N)$ for every configuration, $\Delta\Phi_e = H\Delta x_e$, which vanishes exactly when the update keeps the table or changes it by an integer null vector of H; $x + \Delta x \geq 0$ holds for every update that can occur, because the events it deletes exist. Continuous motion within cells keeps the table as well. For part (i), let each event at z give birth at rate $[\alpha_\tau^\perp(z)]_+$ and die at rate $[-\alpha_\tau^\perp(z)]_+$, independently, as in Section 3.1 with $\alpha_\tau^\perp$ in place of α_τ. A birth or death at $z \in R_m$ changes Φ_m by ± 1, so the mth term of the mean square is $\Delta\tau \sum_i \mathbf{1}_{R_m}(z_i)\,|\alpha_\tau^\perp(z_i)| + o(\Delta\tau)$ given N^τ. By Campbell's formula its expectation is $\Delta\tau \int_{R_m} |\alpha_\tau^\perp|\lambda_\tau \,\mathrm{d}z + o(\Delta\tau)$, which vanishes only if $\alpha_\tau^\perp = 0$ almost everywhere on R_m with respect to $\lambda_\tau \,\mathrm{d}z$. Figure 5a shows the resulting spread of the counts in a simulation.

Proof of Proposition 2(i). Transport confined to cells changes no cell mass, and hence no count of a region, a union of cells. Growth changes the expected count of R_m at rate $\langle \alpha, \mathbf{1}_{R_m}\rangle_\lambda$. The growth fields that keep all counts form the closed subspace $\mathcal{V} = \{a \in L^2(\lambda) : \langle a, \mathbf{1}_{R_m}\rangle_\lambda = 0\ \forall m\}$, the orthogonal complement of the span of the indicators. The orthogonal projection of α onto $\mathcal{V}$ is $\alpha - \sum_m \beta_m \mathbf{1}_{R_m}$ with β solving the normal equations $\Sigma\beta = c$, where $\Sigma_{mn} = \langle \mathbf{1}_{R_m}, \mathbf{1}_{R_n}\rangle_\lambda$ and $c_m = \langle \alpha, \mathbf{1}_{R_m}\rangle_\lambda$. They are solvable even when the indicators are linearly dependent, for example when the total is counted besides the counts of disjoint regions covering the domain: if $\Sigma\chi = 0$, then $\|\sum_m \chi_m \mathbf{1}_{R_m}\|_\lambda^2 = \chi^\top \Sigma \chi = 0$, so $\chi^\top c = \langle \alpha, \sum_m \chi_m \mathbf{1}_{R_m}\rangle_\lambda = 0$, and c lies in the range of the symmetric matrix Σ. Every solution, such as the Moore–Penrose one, gives the same field; disjoint regions make Σ diagonal.

For the dissipation, let $\phi = \log(\lambda/\nu)$, so that $\alpha = -\phi$ and $\alpha^\perp = -\Pi_{\mathcal{V}}\phi$. Formally, for smooth positive solutions with reflecting cell boundaries, $\lambda\, \partial_{\mathbf{n}}\phi = 0$ on every cell face, and integration by parts in each cell gives

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\,\mathrm{GKL}(\lambda_\tau \| \nu) = \int \phi\, \partial_\tau \lambda_\tau \,\mathrm{d}z = -\int \lambda_\tau \|\nabla\phi\|^2 \mathrm{d}z + \langle \phi, \alpha^\perp\rangle_{\lambda_\tau} = -\int \lambda_\tau \|\nabla\phi\|^2 \mathrm{d}z - \|\alpha^\perp\|_{\lambda_\tau}^2.$$

Along a direction with velocity v confined to the cells and growth $a \in \mathcal{V}$, the divergence changes at rate $\langle \nabla\phi, v\rangle_\lambda + \langle \phi, a\rangle_\lambda = \langle \nabla\phi, v\rangle_\lambda + \langle \Pi_{\mathcal{V}}\phi, a\rangle_\lambda$. Among directions of unit WFR norm $\int (\|v\|^2 + a^2)\lambda \,\mathrm{d}z$, the Cauchy–Schwarz inequality makes this most negative for $(v, a) \propto (-\nabla\phi, -\Pi_{\mathcal{V}}\phi)$, the projected flow.

Proof of Proposition 2(ii). Every cell lies inside or outside each region, and regions are unions of cells, so $(Hg)_m = \sum_{j:\, C_j \subseteq R_m} \int_{C_j} \alpha^\perp \lambda \,\mathrm{d}z = \langle \alpha^\perp, \mathbf{1}_{R_m}\rangle_\lambda = 0$; each g_j is finite because $\alpha^\perp \in L^2(\lambda)$ and λ is finite. Thus g lies in the subspace $\ker H$. Every vector of a subspace is a sum of elementary vectors of the subspace, its vectors of minimal support, each conformal to it: their nonzero entries have the signs of the corresponding entries of the vector (Rockafellar, 1969). The elementary vectors of $\ker H$ are the circuits of H, and since H is an integer matrix each is a multiple of a primitive integer vector, so $g = \sum_i c_i u_i$ with $c_i > 0$ and primitive integer circuits u_i conformal to g. These circuits lie in the cone of vectors of $\ker H$ with the sign pattern of g, of dimension at most $J - \operatorname{rank} H$, and Carathéodory's theorem bounds their number by this dimension. The decomposition need not be unique, and a circuit may remove or insert several events at once, as the cubic move of Appendix D.4 does. Figure 5b shows such a decomposition.

Proof of Proposition 2(iii). Write $g_j^\pm = \int_{C_j} [\pm\alpha^\perp]_+ \lambda \,\mathrm{d}z$, so $g_j = g_j^+ - g_j^-$, and let $\pi_j^\pm = [\pm\alpha^\perp]_+ \lambda\, \mathbf{1}_{C_j}/g_j^\pm$, with $\pi_j^\pm = 0$ when $g_j^\pm = 0$. Consider the channels (R_j), which remove one event from C_j and create one there, at total rate $\min(g_j^+, g_j^-)$, and (u_i), which remove $[-u_{ij}]_+$ events from each C_j and create $[u_{ij}]_+$ there, at total rate c_i. A cell with $g_j < 0$ is touched only by circuits with $u_{ij} \leq 0$, so its total removal rate is $g_j^+ + \sum_i c_i |u_{ij}| = g_j^+ + |g_j| = g_j^-$ and its total creation rate is g_j^+; the case $g_j \geq 0$ is symmetric. Removing according to π_j^- and creating according to π_j^+ therefore gives the growth term $\sum_j (g_j^+ \pi_j^+ - g_j^- \pi_j^-) = \alpha^\perp \lambda$. At the sample level, a channel that removes one event is fired by per-capita rates: an event at $z \in C_j$ triggers it at rate $(\text{channel rate}/g_j^-)\,[-\alpha^\perp(z)]_+$, deletes itself and creates the channel's new events from π^+. By Campbell's formula the expected removal and creation intensities are then exactly those above, in the mean-field sense in which branching diffusion realizes equation 1: the fields are evaluated at the

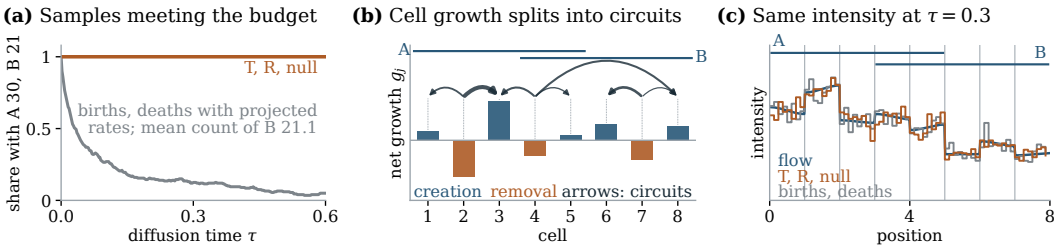


Figure 5: The count-keeping flow on a line of eight unit cells with areas A (cells 1–5) and B (cells 4–8), started from 30 events in A and 21 in B, reference intensity 5, and 200 simulated paths per process. (a) With the projected rates carried out by independent births and deaths, the mean count of B stays near 21 but the share of samples that meet both counts falls to 5%; the pairing of Proposition 2(iii) keeps every sample at A 30, B 21. (b) Net growth g_j of the cells at $\tau = 0.02$ and its conformal decomposition into circuits of H (arrows from removal to creation, width proportional to the rate). (c) At $\tau = 0.3$, both processes follow the intensity of the flow.

intensity of the flow. Every channel removes one event under partition budgets, where the circuits are relocations $e_k - e_j$ between cells of one region, and at a fixed table, where $g = 0$ and only (R_j) remains. A circuit that removes several events at once needs events in several cells at the same time, and its expected rate involves factorial moment measures; replacing them by products of intensities is the mass-action closure, exact in the limit of many events per cell. At small counts it can fail: with cells in A only, in both, in B only and in neither, the counts of A, B and the total fixed at $(1, 1, 2)$, and the single circuit $(-1, 1, -1, 1)$, the table $(0, 1, 0, 1)$ cannot fire it. Every channel changes the table by 0 or by a circuit u_i with $Hu_i = 0$, so the counts stay at b on every path. Conformality is not needed for this bookkeeping, only that the within-cell rates $g_j^- - \sum_i c_i[u_{ij}]_-$ stay nonnegative; conformal circuits maximize pairing within cells and minimize traffic between them.

C Forward Chain, Network and Training Loss (Section 4.2)

The table law of the reference. A Poisson process conditioned on its table places independent events with density proportional to ν in each cell, so $Q^b = \sum_{x \in \mathcal{X}_b} w_Q^b(x)\, Q_x$ with $w_Q^b(x) \propto \mathbf{1}\{Hx = b\} \prod_j \mathrm{Pois}(x_j; \nu(C_j))$. The chain rule of the KL divergence over the table then gives equation 7.

Uniform refresh. With ν constant on a cell C_j of volume $|C_j|$ and mass $\Lambda_j = \int_{C_j} \lambda \, \mathrm{d}z$, refreshing each event at per-capita rate p with a uniform replacement in the cell changes the intensity by $p(\bar{\lambda}_j - \lambda)$, where $\bar{\lambda}_j = \Lambda_j / |C_j|$. Since $\partial_\tau \Lambda_j = p(\Lambda_j - \Lambda_j) = 0$, $\bar{\lambda}_j$ stays fixed. The first variation of $F(\lambda) = \mathrm{GKL}(\bar{\lambda}_j \| \lambda)$ is $1 - \bar{\lambda}_j / \lambda$, so its Fisher–Rao flow $\partial_\tau \lambda = -\lambda \, \delta F / \delta \lambda$ is $\bar{\lambda}_j - \lambda$. Its growth field $a = p(\bar{\lambda}_j / \lambda - 1)$ has $\langle a, 1 \rangle_\lambda = 0$ on the cell, is positive exactly where $\lambda < \bar{\lambda}_j$, and vanishes at $\lambda = \bar{\lambda}_j$. At a fixed table the regions are the cells, and centring $\alpha = -\log(\lambda / \nu)$ within C_j gives $\alpha^\perp = -\log(\lambda / \bar{\lambda}_j) + \mathrm{KL}(\lambda / \Lambda_j \,\|\, \bar{\lambda}_j / \Lambda_j)$ there; both terms differ from $\bar{\lambda}_j / \lambda - 1$ and 0 only at second order in $\lambda / \bar{\lambda}_j - 1$, so $a = p\, \alpha^\perp$ to first order. Over a time s the solution is the arithmetic interpolation $e^{-ps}\lambda + (1 - e^{-ps})\bar{\lambda}_j$, whereas the growth $\alpha^\perp$ alone, $\partial_\tau \log \lambda = -\log(\lambda / \bar{\lambda}_j) + c(\tau)$, gives the geometric interpolation $\lambda^{e^{-s}} \bar{\lambda}_j^{1 - e^{-s}}$ up to normalization; far from $\bar{\lambda}_j$ the refresh death rate is at most p, while $-\log(\lambda / \bar{\lambda}_j)$ is unbounded. Uniform refresh is the only intensity-free delete-and-redraw kernel with this property. If an event at $z \in C_j$ is deleted at rate $p(z)$ and redrawn from a density q on C_j, the intensity changes by $L\lambda = -p\lambda + q\langle p, \lambda \rangle$. Linearized at $\bar{\lambda}_j$, the growth of equation 6 is $\alpha^\perp \lambda = \bar{\lambda}_j - \lambda + O(\|\lambda / \bar{\lambda}_j - 1\|^2)$, and matching $L\lambda = c(\bar{\lambda}_j - \lambda)$ for all λ of mass Λ_j near $\bar{\lambda}_j$ means $(c - p)\delta + q\langle p, \delta \rangle = 0$ for every δ with $\int \delta = 0$. Taking δ concentrated at two points shows that p is constant, then $p = c$, and $L\lambda = c(q\Lambda_j - \lambda) = c(\bar{\lambda}_j - \lambda)$ forces $q = \bar{\lambda}_j / \Lambda_j$, the uniform density. One step of equation 8 realizes the arithmetic solution with $p_k = 1 - e^{-ps}$, and its folded Gaussian step is the heat flow, the transport of equation 6 for constant ν. The two commute: heat flow in the cell keeps the uniform law, and a refresh does not depend on the position it replaces, so applying both over one step in either order gives the mixture equation 8.

Lemma 2 (Forward chain). *Fix a table x. Every step of equation 8 leaves Q_x invariant, so $\mathrm{KL}(\mathrm{Law}(N^k \mid x) \,\|\, Q_x)$ never increases in k, and with $p_K = 1$, $N^K \sim Q_x$ from any start.*

Proof. Condition on x. Every event is updated independently by the kernel $(1 - p_k)\mathcal{G}_{k,j} + p_k \mathcal{U}_j$ of its cell, and no event leaves its cell, so the table is kept. Cells are boxes and the Gaussian step has independent coordinates, so it suffices to show that folding keeps the uniform law on an interval $[0, \ell]$. The fold map is the quotient of the circle $\mathbb{R}/2\ell\mathbb{Z}$ by $s \mapsto -s$. If U is uniform on $[0, \ell]$ and $\varsigma = \pm 1$ is an independent fair sign, ςU is uniform on the circle, and so is $\varsigma U + \sigma\xi$; its fold is therefore uniform on $[0, \ell]$. Because the fold is even and ξ is symmetric, this fold has the law of the fold of $U + \sigma\xi$. Hence $\mathcal{U}_j \mathcal{G}_{k,j} = \mathcal{U}_j$, each event's kernel fixes $\mathcal{U}_j$, and the configuration kernel fixes Q_x. Applying one kernel K with $Q_x \mathsf{K} = Q_x$ to both arguments cannot increase KL divergence (Cover & Thomas, 2006), which gives the monotonicity, in the extended sense when the divergence is infinite. At $p_K = 1$ every event is redrawn from $\mathcal{U}_j$ independently, so $N^K \sim Q_x$. □

The chain forgets the initial positions at a rate set by the refresh probabilities.

Lemma 3 (Loss of initial positions). *Under the conditions of Lemma 2, with $n = \sum_j x_j$ events,*

$$\big\|\mathrm{Law}(N^k \mid N^0) - Q_x\big\|_{\mathrm{TV}} \le \min\Big\{1,\, n \prod_{r=1}^{k} (1 - p_r)\Big\}.$$

Proof. Label events within each cell. Couple the chain started at N^0 with one started from independent uniform positions, using the same perturbations, refresh gates and replacement draws. The latter chain remains cell-uniform. Once a labeled event is refreshed, its two coupled positions coincide thereafter. An event is never refreshed with probability $\prod_{r=1}^{k}(1 - p_r)$; a union bound over n events and the coupling inequality (Levin & Peres, 2017) give the claim, and removing labels cannot increase total variation. □

Set representation. The denoiser combines sinusoidal coordinate features, relative within-cell coordinates, a cell embedding, a diffusion-step embedding and a linear embedding of the full table x. A permutation-equivariant set encoder combines per-event features with pooled configuration features. A linear head returns a displacement in units of the event's cell span. Folding this displacement into the same cell gives the prediction $\widehat{y}_i$ of Section 4.2; no learned deletion or allocation head is required.

More explicitly, a pointwise MLP maps the input features of event i to a hidden vector ψ_i. We concatenate the mean and coordinate-wise maximum of $\{\psi_i\}$ and linearly project them to a global feature $\bar{\psi}$. A second pointwise MLP receives $(\psi_i, \bar{\psi})$ together with the step, table and cell embeddings; a linear displacement head maps its output to three coordinates. Both MLPs use LayerNorm and GELU after each hidden linear layer. The implementation defaults to width 128 and two linear layers per MLP. Coordinate features use sine and cosine frequencies $\{1, 2, 4, 8\}$ in normalized event time and $\{1, 2, 4\}$ on each spatial axis, alongside normalized coordinates and cell-relative coordinates. The mean cell count and the event's cell count divided by $\max(1, \max_j x_j)$ provide two additional scalar features. The full table is normalized by the same maximum before its linear embedding. Empty configurations map to empty predictions.

Training loss. The loss compares the prediction and the clean window cell by cell,

$$\mathcal{L}(\theta) = \mathbb{E} \sum_j \mathcal{D}_\epsilon\big(\widehat{N}^0_\theta|_{C_j},\, N^0|_{C_j}\big), \qquad \mathcal{D}_\epsilon(\zeta, \zeta') = \big[\mathcal{T}_\epsilon(\zeta, \zeta') - \tfrac{1}{2}\mathcal{T}_\epsilon(\zeta, \zeta) - \tfrac{1}{2}\mathcal{T}_\epsilon(\zeta', \zeta')\big]_+, \quad (10)$$

where the expectation is over training windows and a uniform step k, with N^k drawn by equation 8. For equal-sized cell configurations $\zeta = \sum_{i=1}^{n} \delta_{z_i}$ and $\zeta' = \sum_{i'=1}^{n} \delta_{y_{i'}}$, set $c_{ii'} = c(z_i, y_{i'})$ with the scaled squared distance $c(z, y) = (t - t')^2/\omega_t^2 + \|s - s'\|^2/\omega_s^2$ for $z = (s, t)$, $y = (s', t')$. Starting at zero potentials, perform a fixed number of balanced log-domain Sinkhorn iterations,

$$f_i \leftarrow -\epsilon \log \sum_{i'} e^{(g_{i'} - c_{ii'})/\epsilon}, \qquad g_{i'} \leftarrow -\epsilon \log \sum_{i} e^{(f_i - c_{ii'})/\epsilon}, \qquad \Gamma_{ii'} = e^{(f_i + g_{i'} - c_{ii'})/\epsilon}.$$

Define $\mathcal{T}_\epsilon(\zeta, \zeta') = \sum_{ii'} \Gamma_{ii'} c_{ii'}$ after the last iteration and use the clipped self-cost correction $\mathcal{D}_\epsilon$ of equation 10. Training uses 30 iterations and $\epsilon = 0.01$. The objective evaluates the transport

term of the entropic coupling; it does not include the entropy term of the optimized OT value. It is therefore a Sinkhorn-based geometric loss, rather than the full entropic Sinkhorn divergence of Feydy et al. (2019). No metric or statistical-consistency property of that divergence is needed for our conservation results. All atoms carry unit mass, and cell losses are summed per configuration.

D Sampling under a Budget (Section 4.3)

D.1 The reference table law

Given a latent component $\ell \in \{1, \dots, L\}$ with probability π_ℓ, the cell counts are independent Poisson with means $\bar{n}_{\ell j} = \Lambda_\ell p_{\ell j}$, where Λ_ℓ is a total rate and $\sum_j p_{\ell j} = 1$. We fit the parameters by EM on the tables x_i of the training windows, maximizing

$$\sum_i \log \left[\sum_{\ell=1}^{L} \pi_\ell \prod_j \mathrm{Pois}(x_{ij}; \bar{n}_{\ell j}) \right].$$

Responsibilities are proportional to the terms inside the sum, and weighted counts update the rates and cell probabilities. Positive cell pseudo-counts keep every mean positive, and mixture-weight pseudo-counts prevent components receiving zero prior mass. Mixture size and smoothing are selected on training data. Sharing one component across all cells permits dependence between cells after marginalizing ℓ.

For a partition $R_1, \dots, R_M$ covering the domain, sums of independent Poisson counts are Poisson, which gives $\Pr(\ell \mid b) \propto \pi_\ell \prod_m \mathrm{Pois}(b_m; \sum_{C_j \subseteq R_m} \bar{n}_{\ell j})$, and independent Poisson counts conditioned on their sum are multinomial (Kingman, 1993). Conditional on b and a sampled component,

$$(x_j)_{C_j \subseteq R_m} \sim \mathrm{Multinomial}\left(b_m, \left(\frac{\bar{n}_{\ell j}}{\sum_{C_{j'} \subseteq R_m} \bar{n}_{\ell j'}} \right)_{C_j \subseteq R_m} \right), \qquad m = 1, \dots, M,$$

which draws the table exactly from the fitted law restricted to the budget,

$$w_{\mathrm{ref}}(\ell, x \mid b) \propto \mathbf{1}\{Hx = b\}\, \pi_\ell \prod_j \mathrm{Pois}\left(x_j; \bar{n}_{\ell j}\right), \tag{11}$$

and enforces the budget. Placing x_j uniform events in each C_j gives $N^K \sim Q_x$ for the drawn table.

D.2 Reverse steps

The posterior of one event. Fix a cell C, write U for the uniform density on C and $\mathcal{N}_V(\cdot \mid y)$ for the Gaussian density with diagonal covariance V centered at y and folded into C. Folding commutes with symmetric Gaussian steps (proof of Lemma 2), so with $V_k = k\,\sigma_{\mathrm{f}}^2$ and $s_k = \prod_{r \le k}(1 - p_r)$ the forward law of an event with clean position z^0 is

$$\mathrm{Law}(z^k \mid z^0) = s_k\, \mathcal{N}_{V_k}(\cdot \mid z^0) + (1 - s_k)\, U.$$

Bayes' rule with the step kernel $p_k U(z^k) + (1 - p_k)\mathcal{N}_{\sigma_{\mathrm{f}}^2}(z^k \mid z^{k-1})$ gives three branches for z^{k-1} given z^k and z^0, with weights

$$w_{\mathrm{R}} = p_k\, U(z^k), \qquad w_{\mathrm{B}} = (1 - p_k)\, s_{k-1}\, \mathcal{N}_{V_k}(z^k \mid z^0), \qquad w_{\mathrm{F}} = (1 - p_k)(1 - s_{k-1})\, U(z^k).$$

In the refresh branch, $z^{k-1} \sim s_{k-1}\mathcal{N}_{V_{k-1}}(\cdot \mid z^0) + (1 - s_{k-1})U$. In the bridge branch, z^{k-1} is Gaussian with mean $(1-\rho_k)z^k + \rho_k z^0$, $\rho_k = \sigma_{\mathrm{f}}^2/V_k = 1/k$, and variance $\sigma_k^2 = \sigma_{\mathrm{f}}^2 V_{k-1}/V_k$, folded into C. This is exact for the unfolded chain; with folding, the exact bridge is a sum over mirror images of z^0, which we truncate to its leading term away from the walls. In the free branch, the event had been refreshed before and $z^{k-1} \sim \mathcal{N}_{\sigma_{\mathrm{f}}^2}(\cdot \mid z^k)$. The refresh probability is $\pi_k(z^k, z^0) = w_{\mathrm{R}}/(w_{\mathrm{R}} + w_{\mathrm{B}} + w_{\mathrm{F}})$, which grows as $\mathcal{N}_{V_k}(z^k \mid z^0)$ falls. At $k = K$, $p_K = 1$ and every event is refreshed.

Evaluating the posterior at the prediction. Each reverse step evaluates this posterior at the prediction $\widehat{N}^0_\theta$, with the unbalanced coupling Γ of equation 12 in place of the unknown correspondence. For current event i in cell C, the row mass $r_i = \min\{1, \sum_{i'} \Gamma_{ii'}\}$ is the probability that its partner is the barycentric target $\bar{y}_i$; with probability $1 - r_i$ its partner is a predicted event of C that no current event covers, drawn with probability proportional to its unmatched mass $d_{i'} = [1 - \sum_{i''} \Gamma_{i''i'}]_+$. The event is refreshed with the posterior probability averaged over its partner,

$$r_i\, \pi_k(z_i^k, \bar{y}_i) + (1 - r_i) \sum_{i'} \tfrac{d_{i'}}{\sum_{i''} d_{i''}}\, \pi_k(z_i^k, \widehat{y}_{i'}),$$

and its replacement is drawn from the refresh branch around a partner sampled from the same weights, each reweighted by its refresh probability. Otherwise it takes the bridge step of Section 4.3 toward $\bar{y}_i$ or the free step, in the proportions $w_{\mathrm{B}} : w_{\mathrm{F}}$. Every update stays in the event's cell, so the table is kept exactly. The forward scale σ_{f} is 0.1 times each cell's span on every coordinate, and $p_k = k/K$.

Sampling coupling.

$$\begin{aligned}\Gamma = \arg\min_{\Gamma \geq 0} \langle \Gamma, c\rangle + \epsilon_{\mathrm{s}}\, \mathrm{KL}(\Gamma \| \mathbf{1}\mathbf{1}^\top) \\ + \gamma\big[\,\mathrm{KL}(\Gamma\mathbf{1}\|\mathbf{1}) + \mathrm{KL}(\Gamma^\top\mathbf{1}\|\mathbf{1})\big],\end{aligned} \tag{12}$$

with $c_{ii'} = c(z_i, \widehat{y}_{i'})$. For nonnegative arrays, $\mathrm{KL}(a\|b) = \sum_i [a_i \log(a_i/b_i) - a_i + b_i]$. The coupling equation 12 uses unit source and target masses, $\epsilon_{\mathrm{s}} = 0.1$, $\gamma = 90$, and coordinate scales $\omega_t = 0.1T$ and $\omega_s = 0.1$ times the spatial span. Row masses are protected by a small numerical denominator when forming barycentres. Partition-budget runs solve this objective with a float64 CPU solver.

D.3 Allocation and position errors

Our sampler draws a final table and positions given it, so its law has the form of equation 3, $P^b_\theta = \sum_x w^b_{\mathrm{alg}}(x)\, P_\theta(\cdot \mid x)$, with final-table law w^b_{alg} and output law $P_\theta(\cdot \mid x)$ given that table. Its error splits exactly.

Proposition 3 (Allocation and position errors)**.** *In $[0, \infty]$, without further assumptions,*

$$\mathrm{KL}\left(P^b_\star \,\|\, P^b_\theta\right) = \mathrm{KL}\left(w^b_\star \,\|\, w^b_{\mathrm{alg}}\right) + \mathbb{E}_{x \sim w^b_\star}\, \mathrm{KL}\left(P_{\mathrm{data}}(\cdot \mid X = x) \,\|\, P_\theta(\cdot \mid x)\right). \tag{13}$$

The reverse updates (T) and (R) keep the table, so they act on the second term only; under Joint budgets the first term vanishes for every method.

Proof of Proposition 3. Let $P = P^b_\star$ and $\tilde{P} = P^b_\theta$. Every state of our sampler lies in $\mathcal{F}_b$ (Section 4), so both laws are concentrated on $\mathcal{F}_b$, the disjoint union of the slices $\mathcal{F}_x = \{N : x(N) = x\}$, $x \in \mathcal{X}_b$, and the table $X = x(N)$ takes countably many values. Given $X = x$, P has the conditional law $P_x = P_{\mathrm{data}}(\cdot \mid X = x)$, since $\{X = x\} \subseteq \{\Phi(N) = b\}$, and $\tilde{P}$ has $\tilde{P}_x = P_\theta(\cdot \mid x)$; for tables with $w^b_{\mathrm{alg}}(x) = 0$ let $\tilde{P}_x$ be any law on $\mathcal{F}_x$, which changes neither side. Write $w_\star = w^b_\star$ and $w = w^b_{\mathrm{alg}}$. If $w_\star(x) > 0 = w(x)$ for some x, then $P(\mathcal{F}_x) > 0 = \tilde{P}(\mathcal{F}_x)$ and both sides are infinite. Otherwise $w_\star \ll w$. If $P_x \not\ll \tilde{P}_x$ for some x with $w_\star(x) > 0$, the right side is infinite, and so is the left, since a set $E \subseteq \mathcal{F}_x$ with $\tilde{P}_x(E) = 0 < P_x(E)$ has $\tilde{P}(E) = 0 < P(E)$. If $P_x \ll \tilde{P}_x$ for every such x, then $P \ll \tilde{P}$, and on $\mathcal{F}_x$ the density is $\frac{\mathrm{d}P}{\mathrm{d}\tilde{P}} = \frac{w_\star(x)}{w(x)}\, \frac{\mathrm{d}P_x}{\mathrm{d}\tilde{P}_x}$. Integrating its logarithm against P over $\mathcal{F}_x$ gives $w_\star(x) \log \frac{w_\star(x)}{w(x)} + w_\star(x)\, \mathrm{KL}(P_x \| \tilde{P}_x)$. The second term is nonnegative, and the first is at least $w_\star(x) - w(x)$, whose sum over x converges absolutely, so the sum over x may be split term by term. Summing gives equation 13 in $[0, \infty]$; no absolute continuity is assumed, and when it fails both sides are infinite. Nonnegativity of the second term gives $\mathrm{KL}(P^b_\star \| P^b_\theta) \geq \mathrm{KL}(w^b_\star \| w^b_{\mathrm{alg}})$, and the same decomposition gives $\mathrm{TV}(w_\star, w) \leq \mathrm{TV}(P, \tilde{P}) \leq \mathrm{TV}(w_\star, w) + \mathbb{E}_{x \sim w_\star}\, \mathrm{TV}(P_x, \tilde{P}_x)$, which is always finite. Under partition budgets every update keeps the initial table, so $w^b_{\mathrm{alg}} = w_{\mathrm{ref}}(\cdot \mid b)$. Under overlap, w^b_{alg} is the actual law of the final table for the chosen network and proposal schedule; it may depend on θ through the fixed proposal weights, as well as on the initialization and

the finite number of updates, while conditional on the weights each subsequent table kernel keeps the reference invariant. The chain rule applies to the actual final-table law however it is generated. The position law given the final table may depend on the budget and on the whole table path, and equation 13 uses only its conditional law given the final table.

D.4 A CHALLENGING CASE: OVERLAPPING BUDGETS

Regions and initialization. Coverage areas are unions of spatial grid cells S_g whose centers lie within a base's radius; with the daily windows they define H. Only spatial cells that contain training events are admitted, and the reference and all tables live on these cells. Both coverage budgets include the daily totals, so the total count n is fixed as well. We first draw a nonnegative provisional table x' that respects the daily totals, then solve

$$\min_{x \in \mathbb{Z}_+^J} \|x - x'\|_1 \quad \text{subject to } Hx = b$$

with zero upper bounds on excluded cells. This integer program is solved once per sample.

Reference. The reference is the Poisson mixture of Appendix D.1, fitted on training weeks over the admitted spatial cells g, with total rates Λ_ℓ and cell probabilities $p_{\ell g}$; five-fold training likelihood selects four components on the wildfire data. Let $P_\ell = \sum_g p_{\ell g}$ over admitted cells, and $x_g = \sum_v x_{gv}$ for the table $x = (x_{gv})$ over cells and days. Given ℓ, the count of cell g on day v is Poisson with mean $\Lambda_\ell p_{\ell g}$ times a day weight that is shared by all components and sums to one over the days. On the fiber $\mathcal{X}_b$, the set of feasible tables, the total and the daily totals are fixed, so the day weights cancel and equation 11 becomes

$$w_{\mathrm{ref}}(\ell, x \mid b) \propto \mathbf{1}\{Hx = b\}\ \pi_\ell\ \mathrm{Pois}(n; \Lambda_\ell P_\ell) \prod_g \Big(\frac{p_{\ell g}}{P_\ell}\Big)^{x_g} \prod_{g,v} \frac{1}{x_{gv}!}.$$

The Gibbs step draws ℓ with probability proportional to $\pi_\ell \,\mathrm{Pois}(n; \Lambda_\ell P_\ell) \prod_g (p_{\ell g}/P_\ell)^{x_g}$. For fixed ℓ, only $p_{\ell g}$ and the factorials enter the ratio in equation 14, because every relocation keeps its event's day. At initialization, a week with n events draws ℓ with probability proportional to $\pi_\ell \,\mathrm{Pois}(n; \Lambda_\ell P_\ell)$, and the provisional table distributes each daily total over the admitted cells multinomially with probabilities $p_{\ell g}/P_\ell$. Events are placed uniformly in their cells, at initialization and after every relocation.

Table updates. Before each reverse step, the sampler redraws ℓ from $w_{\mathrm{ref}}(\ell \mid x, b)$ and attempts null moves $x \to x' = x + \Delta x$, each accepted with the Metropolis–Hastings probability

$$\mathrm{acc}(x, x') = \min\big\{1,\ w_{\mathrm{ref}}(\ell, x' \mid b)\, q(x' \to x) \,/\, [w_{\mathrm{ref}}(\ell, x \mid b)\, q(x \to x')]\big\}, \tag{14}$$

where q is the proposal probability below.

Proposition 4 (Table updates)**.** *With fixed proposal weights, redrawing ℓ from $w_{\mathrm{ref}}(\ell \mid x, b)$ and then applying equation 14 keeps $Hx = b$ and leaves $w_{\mathrm{ref}}(\ell, x \mid b)$ invariant. If $\mathcal{X}_b$ is finite, $w_{\mathrm{ref}} > 0$ on it and the proposed moves connect $\mathcal{X}_b$ in both directions, the lazy update converges to $w_{\mathrm{ref}}(\ell, x \mid b)$ geometrically fast from every feasible start.*

Positive Poisson means make w_{ref} positive, and the Markov-basis moves below connect $\mathcal{X}_b$, so the proposition applies to our sampler; we prove it at the end of this subsection.

Guided paired proposal. Let $\mathrm{sig}(g) = (\mathbf{1}\{S_g \subseteq D_a\})_a$, over the coverage areas D_a, be the coverage signature of spatial cell g, and write a table entry as $u = (g, v)$ with day v. The coupling Γ of the first reverse step fixes, for the rest of the sample, positive weights

$$d_u^{\mathrm{out}} = \varepsilon_d + \sum_{i:\, z_i \in u} \Big[1 - \sum_{i'} \Gamma_{ii'}\Big]_+, \qquad d_u^{\mathrm{in}} = \varepsilon_d + \sum_{i':\, \widehat{y}_{i'} \in u} \Big[1 - \sum_{i} \Gamma_{ii'}\Big]_+, \qquad \varepsilon_d = 0.05,$$

the current mass that the prediction leaves unexplained and the predicted mass that no current event covers, so that proposals follow the signs of the net growth implied by the prediction (Proposition 2(ii)); the null moves of the first reverse step, made before that coupling, use the same proposal with $d^{\mathrm{out}} = d^{\mathrm{in}} = 1$. The prediction has the same table as the current configuration, so

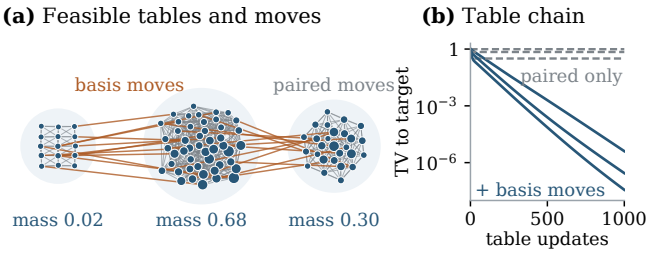


Figure 6: Table chain on a three-area budget (two admitted cells per signature, every area count four, total six). (a) The 90 feasible tables, sized by target probability. Paired moves (grey) never change the number of events per signature and split the tables into three classes; the cubic basis move (rust, 12 of its edges drawn per pair of classes) joins them. (b) Exact total-variation distance to the target from the most probable table of each class: paired moves stall at one minus the mass of the starting class, and mixing in basis moves with probability 0.1 gives geometric convergence.

a guide toward the predicted table would carry no information, whereas the unbalanced coupling across cells does. A proposal picks an occupied source entry u_1 with probability $d^{\text{out}}_{u_1}/Z_1(x)$, $Z_1(x) = \sum_{u:\, x_u>0} d^{\text{out}}_u$; a destination $u_2 = (g', v_{u_1})$ with $g' \neq g_{u_1}$ admitted, with probability $d^{\text{in}}_{u_2}/Z_2(u_1)$; and a compensating relocation $u_3 \to u_4$ with probability $d^{\text{out}}_{u_3} d^{\text{in}}_{u_4}/Z_3(x, u_1, u_2)$ from the set of relocations between distinct spatial cells whose signature change is opposite to that of $u_1 \to u_2$, that keep their day, whose source holds an available event ($x_{u_3} > \mathbf{1}\{u_3 = u_1\}$) and, under daily area counts, that use the day of u_1; Z_2 and Z_3 normalize over these sets. The tuple $t = (u_1, u_2, u_3, u_4)$ changes the table by $\Delta(t) = -\mathbf{1}_{u_1} + \mathbf{1}_{u_2} - \mathbf{1}_{u_3} + \mathbf{1}_{u_4}$ with $H\Delta(t) = 0$, and has probability

$$p(t \mid x) = \frac{d^{\text{out}}_{u_1}}{Z_1(x)} \frac{d^{\text{in}}_{u_2}}{Z_2(u_1)} \frac{d^{\text{out}}_{u_3} d^{\text{in}}_{u_4}}{Z_3(x, u_1, u_2)}.$$

Each relocation keeps its event's day, which preserves the daily totals, and the opposite signature changes preserve the area counts. Entries may coincide, so the net change has four, three or two nonzero entries, or none.

Proposal probability and acceptance. For $y \neq x$, the proposal probability in equation 14 is $q(x \to y) = \sum_{t:\, \Delta(t)=y-x} p(t \mid x)$, summed over every tuple with that net change: for four distinct entries both pairings and both orders, for three entries both orders, and for two entries either the repeated relocation or both orders through every admissible intermediate entry. The remaining probability, from empty choice sets and zero net changes, is $q(x \to x)$; an empty table is kept. The reverse probability $q(y \to x)$ is computed from the proposed table with the same fixed weights, and the acceptance uses both in log space. Conditional on the weights, the table kernel depends only on the table and the component, so the argument of Proposition 4 applies; the weights and the start table may be dependent, which is part of the initialization. Weights updated from the current configuration are not covered by this argument, since a kernel chosen by the current state need not keep the target that each fixed kernel keeps. At reverse step k the sampler makes $\text{round}(c\, n\, k / \sum_{r \le K} r)$ attempts, with $c = 30$ under weekly and $c = 100$ under daily area counts. These attempts are concentrated at early reverse steps, when the training corruption is strongest. Over the 156 test weeks with five samples each, the chain accepted 8.4 moves per event under weekly and 4.2 under daily area counts, 28% and 4.2% of the attempts, and at least 168 and 41 moves in every sampled week.

Checks. We checked the implementation on small fibers with five cells, two days and two overlapping areas. Enumerating every outcome of the procedure reproduces the implemented proposal probabilities to 10^{-16}, and the resulting transition matrix M leaves the target π invariant up to rounding: the entries of $\pi\mathsf{M} - \pi$ are at floating-point precision, below the threshold of 10^{-12} fixed before running, on fibers of 5, 21 and 34 tables, which are all the feasible tables of these budgets. The same acceptance without the proposal ratio leaves residuals between 2×10^{-3} and 1.3×10^{-2}. These checks concern the implemented proposal probabilities and acceptance.

When paired moves connect the feasible set. The paired moves form a library of feasible integer moves in the null space of H, but they need not form a Markov basis, which would connect

every feasible set (Diaconis & Sturmfels, 1998). A relocation changes the counts only through the signatures of its source and destination, so a paired move keeps the multiset of signatures up to replacing two signatures σ_1, σ_3 by two others σ_2, σ_4 with $\sigma_2 + \sigma_4 = \sigma_1 + \sigma_3$. Consider three areas on one day and admitted cells whose signatures are only 000, 110, 011, 101 and 111. The tables with one event in each of 110, 011, 101 and those with two events in 111 and one in 000 meet the same budget: every area count is two and the total is three. From the first, any two sources sum to 121, 211 or 112, and no other pair of admitted signatures has these sums, so every paired move keeps the signature multiset and the second table is never reached. Mixing moves from a Markov basis of H into the proposals, with any positive probability, removes such gaps; by Proposition 4 the chain then converges to $w_{\mathrm{ref}}(\ell, x \mid b)$ from any feasible start. Figure 6 computes this by enumeration for the same signatures with two admitted cells each, every area count four and total six (90 tables), a product-Poisson table model, uniform proposal weights and the lazy update. Paired moves keep the chain in the class of tables with its starting signature counts, three classes of target mass 0.02, 0.68 and 0.30, so from the most probable table of each class its distance stalls at one minus that mass; with the basis moves above, which here are the single relocations and the cubic move $110 + 011 + 101 \leftrightarrow 111 + 111 + 000$, at $\varepsilon_{\mathcal{M}} = 0.1$, it decays geometrically from every start.

Markov-basis proposal. Because a relocation affects the counts only through signatures and days, connectivity reduces to the aggregated table $y_{\sigma v}$, the number of events with signature σ on day v. Columns of H with the same signature and day are equal, so single relocations between cells of equal signature and day, together with the lifts of a Markov basis $\mathcal{M}$ of the aggregated constraint matrix, whose columns are the admitted pairs (σ, v) and whose rows are the area counts and daily totals, form a Markov basis of H, as holds for matrices with repeated columns (Aoki et al., 2012): tables with the same y are joined by single relocations, and along a path of aggregated tables that $\mathcal{M}$ connects, every move lifts to a cell move, because an aggregated table that stays nonnegative after a move holds enough events in each pair (σ, v) from which the move removes events. We compute $\mathcal{M}$ once per coverage structure (Diaconis & Sturmfels, 1998). With probability $\varepsilon_{\mathcal{M}}$, fixed in advance, a proposal is a basis move. With probability $\frac{1}{2}$ it is a single relocation from a uniformly chosen occupied cell to a uniformly chosen other admitted cell of the same signature and day; otherwise it draws $m \in \mathcal{M}$ and a sign uniformly, removes one event from a uniformly chosen occupied cell of each pair (σ, v) with $m_{\sigma v} < 0$, once per unit, and adds events to uniformly chosen admitted cells of the pairs with $m_{\sigma v} > 0$; it is rejected if some pair lacks events. Its proposal probability is a product of these uniform choices, summed over the orders that give the same table, and the reverse move is the reverse relocation or uses the opposite sign, so equation 14 applies to the mixture of proposal types with fixed mixing probabilities. The basis moves alone connect $\mathcal{X}_b$, since $\mathcal{M}$ connects every aggregated fiber, and so does the mixture.

Proof of Proposition 4. Accepted increments satisfy $H\Delta x = 0$, and proposals leaving $\mathbb{Z}_+^J$ are rejected, so every visited table lies in $\mathcal{X}_b$. Write $w(\ell, x) = w_{\mathrm{ref}}(\ell, x \mid b)$. For fixed ℓ and fixed proposal weights, the step equation 14 is a Metropolis–Hastings kernel M_ℓ for $w(\ell, \cdot)$: for $x' \neq x$, $w(\ell, x)\, q(x \to x')\, \mathrm{acc}(x, x') = \min\{w(\ell, x) q(x \to x'),\, w(\ell, x') q(x' \to x)\}$ is symmetric in x, x', so $\sum_x w(\ell, x) \mathsf{M}_\ell(x, x') = w(\ell, x')$ (Hastings, 1970). The full update draws $\ell' \sim w(\ell' \mid x)$ and then $x' \sim \mathsf{M}_{\ell'}(x, \cdot)$, hence

$$\sum_{\ell, x} w(\ell, x)\, w(\ell' \mid x)\, \mathsf{M}_{\ell'}(x, x') = \sum_x w(\ell', x)\, \mathsf{M}_{\ell'}(x, x') = w(\ell', x').$$

Summing over ℓ' gives the invariance of the table marginal (Tierney, 1994); the same holds for a mixture of proposal types with fixed mixing probabilities, each accepted by equation 14 with its own proposal probabilities.

For convergence, the state space $\{1, \ldots, L\} \times \mathcal{X}_b$ is finite by assumption (every count bounds the cells of its region, and the daily totals put every admitted cell in some region), and w is positive on it by assumption, as the mixture weights and the smoothed Poisson means make it. The Gibbs step reaches every component with positive probability. For fixed ℓ, a move $x \to x'$ that is proposed with positive probability has a reverse that is also proposed with positive probability: for paired moves, the reverse tuple uses the same entries in the opposite roles and every weight is at least $\varepsilon_d > 0$, and a basis move is reversed by the opposite sign or the reverse relocation; its acceptance is therefore positive. If these moves connect $\mathcal{X}_b$, the chain is irreducible, and holding with probability $\frac{1}{2}$ makes it aperiodic while keeping w invariant. An irreducible aperiodic chain on a finite set converges to its

unique invariant law at a geometric rate (Levin & Peres, 2017, Theorem 4.9). The sampler makes finitely many updates, so its final table follows this law only approximately; this error is counted in the table term of equation 13.

D.5 The sampling algorithm

Algorithm 1 lists the steps of the sampler under partition and overlapping budgets.

Algorithm 1 Sampling under a count budget b

if the regions form a partition **then**
  draw ℓ and the table x exactly from $w_{\mathrm{ref}}(\cdot \mid b)$ (Appendix D.1)
else
  draw a provisional table under the daily totals; find the nearest feasible table x by the integer program of Appendix D.4; set the proposal weights $d^{\mathrm{out}} = d^{\mathrm{in}} = 1$
end if
place x_j uniform events in every cell C_j: $N^K \sim Q_x$
for $k = K, \dots, 1$ **do**
  if the budget is overlapping **then**
    redraw ℓ given x; attempt null moves with the current weights, accept by equation 14, and redraw moved events uniformly in their new cells (null moves)
  end if
  predict $\widehat{N}_\theta^0$ with the current x (Section 4.2); couple current and predicted events by equation 12
  if the budget is overlapping and $k = K$ **then**
    fix $d^{\mathrm{out}}, d^{\mathrm{in}}$ from this coupling for all later steps (Appendix D.4)
  end if
  for every event, draw the posterior branch (Appendix D.2): refresh near its partner (R)
    or move along the bridge toward $\bar{y}_i$ or take the free step (T)
  call the result N^{k-1} and **assert** $\Phi(N^{k-1}) = b$
end for
return N^0

E Experimental Details and Additional Results (Section 5)

E.1 Setup

Budgets. Partition budgets are defined on $8 \times 8 \times 8$ equal space–time cells (time by the two spatial axes).

Metrics. MMD compares per-window features with a Gaussian kernel whose bandwidth is the median pairwise distance of the pooled features: the event count divided by the largest count in the comparison, normalized 16-bin histograms of event times, of gaps between consecutive events and of each spatial coordinate, and the normalized 8×8 histogram of spatial cell counts; we report the square root of the plug-in estimate. CMMD applies the same statistic to within-cell features, the normalized cell count and 8-bin histograms of time and of each coordinate relative to the cell, in every budget cell over the windows whose budget puts at least one event there, and averages over these cells. STSD treats each window as a probability measure with equal weights on its events, so the mass term of the unbalanced divergence vanishes, and uses the debiased unbalanced Sinkhorn divergence with squared Euclidean cost on coordinates scaled by a tenth of the time span and of the larger spatial span, entropic regularization 0.01 and marginal penalty 0.9; with $S(\mu, \nu) = \mathrm{UOT}(\mu, \nu) - \frac{1}{2}\mathrm{UOT}(\mu, \mu) - \frac{1}{2}\mathrm{UOT}(\nu, \nu)$ for the unbalanced transport value UOT of two windows, $\mathrm{STSD} = \frac{1}{n}\sum_i S(\mu_i, \nu_{\pi(i)})$, where π is the minimum-cost assignment of generated windows μ_i to real windows ν_j.

Training. The networks for the partition datasets use $K = 10$ diffusion steps and are trained with AdamW at learning rate 3×10^{-4}, with 500 warmup steps, cosine decay to 10^{-5}, weight decay 0.01 and batch size 32, for 67 epochs, or 64 on EMS; the wildfire network is trained for 320 epochs.

Baselines. DSTPP keeps its published architecture and 500 diffusion steps and is trained with AdamW at learning rate 10^{-3}, with warmup and linear decay, and batch size 64, for up to 2000 epochs with validation every 10 epochs. The branching-diffusion baselines use the same set-network width and the same $K = 10$ steps as ours and are trained with Adam at learning rate 10^{-4} and batch size 32 for the same number of epochs as ours. BD + Count Penalty tilts the keep and birth probabilities of each branching-diffusion reverse step by minimizing their KL divergence from the original probabilities plus $c_{\text{pen}} \sum_m (\bar{\Phi}_m - b_m)^2$, where $\bar{\Phi}_m$ is the expected count of region R_m after the step; c_{pen} is tuned per setting on a development split, between 0.1 and 10, and no repair follows.

Every trained method is run with five training seeds and Uniform-Budget with five sampling repeats; tables report means and standard deviations over the runs. Diversity, the mean distance between samples for the same budget, has no fixed direction without a real conditional reference and is not ranked.

E.2 Partition budgets

Table 1 summarizes all sixteen settings and Table 2 lists every value.

Table 1: Summary over the sixteen partition-budget settings: geometric mean of the ratio to Uniform-Budget, the range of budget violation, and the number of (setting, metric) pairs, out of 48, with the lowest mean among exactly feasible methods; Div. is not ranked. †Violates the budget; excluded from ranking.

Method	Budget violated	Geometric mean of ratio to Uniform-Budget MMD↓	CMMD↓	STSD↓	Div.	Lowest means
Uniform-Budget	0	1	1	1	1	0/48
DSTPP + Repair	0	0.43	0.50	0.78	0.88	0/48
BD + Repair	0	0.91	0.93	0.83	0.69	0/48
BD + Step Repair	0	0.93	0.98	0.78	0.64	0/48
BD + Count Penalty†	68–100%	1.59	1.47	0.86	0.52	–
Ours	0	**0.14**	**0.29**	**0.71**	0.77	48/48

Samples over time. Figure 7 shows the Earthquake samples of Figure 3a over time. Real events arrive evenly over the 30-day window, with every two-day rate within 5% of the mean, and ours stay within 12%. Repaired DSTPP piles events toward the end of the window, up to 44% above the mean, and the branching-diffusion baselines fall 12–36% below the mean in the first and last two days.

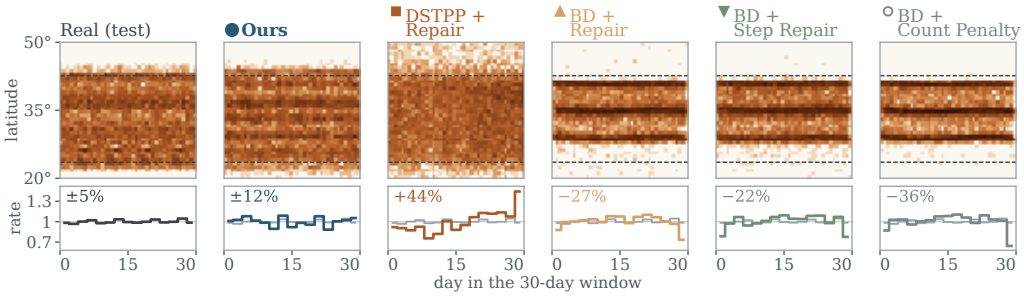


Figure 7: The Earthquake samples of Figure 3a over time. Top: events by day and latitude, pooled over the 150 test windows; dashed lines hold the central 90% of real events. Bottom: events per two days relative to the window mean (grey: real data) and its largest deviation.

Table 2: All partition-budget test values: means over runs, with standard deviations in small type. Tint bins by ratio to the lowest exactly-feasible mean of the block (≤ 1, ≤ 1.05, ≤ 1.25, ≤ 2); bold marks the lowest mean. Diversity is not ranked. †Violates the budget in 68–100% of samples.

	Syn-1				Syn-2			
Method	MMD	CMMD	STSD	Div.	MMD	CMMD	STSD	Div.
Total (1 cell)								
Uniform-Budget	.349 ±.002	.455 ±.004	1.26 ±<.001	1.38	.343 ±.003	.454 ±.004	1.30 ±.001	1.42
DSTPP + Repair	.126 ±.038	.134 ±.038	.899 ±.006	1.08	.132 ±.023	.143 ±.027	.917 ±.005	1.27
BD + Repair	.277 ±.032	.250 ±.038	1.01 ±.019	1.11	.254 ±.018	.289 ±.027	1.22 ±.027	1.32
BD + Step Repair	.405 ±.011	.434 ±.016	.933 ±.008	.931	.515 ±.037	.611 ±.029	1.10 ±.029	.871
BD + Count Penalty†	.623	.709	.871	.552	.785	.908	1.17	.443
Ours	**.033** ±.002	**.034** ±.001	**.880** ±.001	1.07	**.031** ±.002	**.031** ±.002	**.873** ±.001	1.15
Temporal (8 cells)								
Uniform-Budget	.343 ±.002	.178 ±<.001	1.26 ±<.001	1.35	.329 ±.002	.191 ±.001	1.29 ±.002	1.39
DSTPP + Repair	.085 ±.009	.061 ±.006	.941 ±.004	1.13	.098 ±.007	.070 ±.006	.968 ±.005	1.28
BD + Repair	.240 ±.026	.157 ±.008	1.02 ±.018	1.12	.238 ±.013	.162 ±.012	1.21 ±.023	1.30
BD + Step Repair	.314 ±.008	.171 ±.003	.962 ±.011	1.00	.345 ±.039	.192 ±.021	1.08 ±.016	.992
BD + Count Penalty†	.539	.342	.861	.557	.629	.382	1.04	.545
Ours	**.034** ±.002	**.038** ±.001	**.880** ±.001	1.01	**.031** ±.002	**.037** ±.001	**.871** ±.002	1.05
Spatial (64 cells)								
Uniform-Budget	.151 ±.002	.178 ±.001	1.13 ±.003	1.15	.148 ±.002	.153 ±.002	1.06 ±.002	1.11
DSTPP + Repair	.097 ±.011	.129 ±.002	.985 ±.004	1.08	.128 ±.008	.129 ±.002	1.02 ±.003	1.10
BD + Repair	.158 ±.007	.170 ±.003	1.04 ±.007	1.08	.146 ±.007	.158 ±.001	1.07 ±.010	1.09
BD + Step Repair	.149 ±.011	.161 ±.002	1.02 ±.007	1.03	.119 ±.004	.146 ±.002	.995 ±.003	1.00
BD + Count Penalty†	.337	.298	.928	.800	.327	.300	.985	.804
Ours	**.025** ±.003	**.080** ±.001	**.864** ±.002	.904	**.023** ±.002	**.086** ±.001	**.857** ±.002	.911
Joint (512 cells)								
Uniform-Budget	.131 ±.002	.276 ±.002	.551 ±.003	.550	.112 ±.002	.233 ±<.001	.548 ±.004	.548
DSTPP + Repair	.105 ±.002	.271 ±.001	.542 ±.001	.546	.102 ±.002	.232 ±.001	.544 ±.001	.547
BD + Repair	.130 ±.003	.277 ±.002	.546 ±.001	.536	.111 ±.002	.233 ±.002	.545 ±.002	.545
BD + Step Repair	.132 ±.006	.276 ±.001	.518 ±.001	.477	.103 ±.008	.235 ±.002	.509 ±.002	.464
BD + Count Penalty†	.537	.490	.822	.653	.480	.469	.853	.697
Ours	**.023** ±.003	**.257** ±.001	**.505** ±.001	.503	**.021** ±.002	**.217** ±.002	**.504** ±.002	.500

	Earthquake				EMS			
Method	MMD	CMMD	STSD	Div.	MMD	CMMD	STSD	Div.
Total (1 cell)								
Uniform-Budget	.659 ±.001	.752 ±.002	1.27 ±.007	1.23	.787 ±.001	.870 ±.001	.851 ±.003	.736
DSTPP + Repair	.294 ±.023	.308 ±.042	.825 ±.028	.988	.123 ±.016	.123 ±.019	.434 ±.006	.493
BD + Repair	.705 ±.037	.662 ±.017	.764 ±.006	.369	.895 ±.043	.953 ±.024	.499 ±.011	.233
BD + Step Repair	.692 ±.039	.648 ±.011	.754 ±.006	.390	.849 ±.050	.907 ±.026	.469 ±.007	.261
BD + Count Penalty†	.725	.706	.797	.334	.976	1.00	.616	.157
Ours	**.213** ±.008	**.215** ±.011	**.708** ±.001	.772	**.054** ±.003	**.045** ±.002	**.420** ±<.001	.471
Temporal (8 cells)								
Uniform-Budget	.652 ±.002	.413 ±.004	1.27 ±.005	1.20	.777 ±.001	.533 ±.001	.829 ±.001	.711
DSTPP + Repair	.334 ±.030	.173 ±.020	.865 ±.031	1.00	.145 ±.017	.080 ±.005	.448 ±.006	.505
BD + Repair	.691 ±.042	.331 ±.008	.767 ±.005	.354	.884 ±.051	.639 ±.066	.484 ±.014	.260
BD + Step Repair	.665 ±.045	.307 ±.010	.753 ±.004	.390	.799 ±.059	.515 ±.050	.456 ±.007	.316
BD + Count Penalty†	.700	.375	.794	.321	.964	.800	.578	.174
Ours	**.202** ±.009	**.114** ±.005	**.711** ±.003	.736	**.047** ±.003	**.036** ±.001	**.419** ±<.001	.458
Spatial (64 cells)								
Uniform-Budget	.321 ±.003	.277 ±.002	.766 ±.003	.782	.497 ±.002	.262 ±.001	.506 ±<.001	.507
DSTPP + Repair	.212 ±.015	.247 ±.003	.738 ±.003	.767	.106 ±.014	.204 ±.004	.437 ±.002	.466
BD + Repair	.367 ±.051	.320 ±.006	.747 ±.006	.701	.361 ±.027	.275 ±.006	.449 ±.002	.411
BD + Step Repair	.325 ±.022	.291 ±.003	.709 ±.004	.655	.257 ±.010	.247 ±.007	.436 ±.001	.423
BD + Count Penalty†	.316	.335	.702	.622	.434	.297	.445	.364
Ours	**.115** ±.011	**.198** ±.004	**.681** ±.001	.674	**.039** ±.001	**.103** ±.003	**.421** ±<.001	.440
Joint (512 cells)								
Uniform-Budget	.318 ±.004	.329 ±.002	.508 ±.002	.467	.458 ±.001	.222 ±.001	.342 ±.001	.327
DSTPP + Repair	.272 ±.014	.324 ±.002	.493 ±.003	.464	.159	.206	.308	.311
BD + Repair	.360 ±.058	.347 ±.004	.497 ±.010	.405	.250 ±.051	.222 ±.003	.304 ±.004	.286
BD + Step Repair	.322 ±.021	.349 ±.003	.444 ±.002	.343	.189 ±.019	.214 ±.003	.293 ±.002	.272
BD + Count Penalty†	.380	.409	.512	.408	.374	.287	.313	.279
Ours	**.108** ±.006	**.289** ±.004	**.403** ±.002	.371	**.035** ±.003	**.143** ±.001	**.270** ±<.001	.267

E.3 WILDFIRE RESPONSE: OVERLAPPING COVERAGE BUDGETS

Data. Ignitions come from the 7th edition of the Fire Program Analysis fire-occurrence database (Short, 2026), restricted to California. Each record gives a discovery date, a discovery time and a location; records without a time or a location, or outside the state's bounding box, are dropped. Locations are projected to kilometers by an equirectangular approximation around $(37°\mathrm{N}, 119.5°\mathrm{W})$. Windows are consecutive weeks from 1 January 2001 to 30 December 2024, with event times in hours $t \in [0, 168)$; weeks with 50 to 500 ignitions are used. Consecutive blocks of 16 weeks are assigned to training, validation and test in the ratio 3:1:1, dropping the first and last week of each block; after the count filter this leaves 480 training, 136 validation and 156 test weeks. Base locations come from the CAL FIRE facilities layer (California Department of Forestry and Fire Protection, 2026): 16 air-attack bases and 17 helitack bases. The radii, 124 km for air tankers and 60 km for helicopters, are proxies for initial-attack reach, not dispatch rules.

Protocol. The spatial grid has 16×16 cells and seven daily windows. Besides the two coverage budgets of Section 5.3, Table 3 also reports weekly counts for the air-attack areas alone. Every method uses only the budget of a test week. We correct DSTPP because its repaired samples had lower MMD and CMMD than the better repaired branching-diffusion baseline in most partition settings (Table 2). DSTPP is trained on the training weeks without budgets; its samples are repaired or projected afterwards. Local repair applies the region-wise deletion and insertion of Section 3.3 to the areas in turn and repeats the pass 100 times. Global projection first matches the daily totals by one-dimensional minimal transport between days, then finds the feasible spatial table closest to the sample in total transport distance, an integer program over the admitted cells, and realizes each flow by moving the events of the source cell nearest to the destination. Moved events are placed uniformly in their destination cell and day; a relative-placement variant keeps their position relative to the cell and their time of day. Distances are paired: each sample is compared with the actual week of its budget by the spatio-temporal Sinkhorn divergence of Section 5.1. CMMD uses the model's $7 \times 16 \times 16$ grid. Planning errors use continuous disks around air-attack bases: weekly counts within 90 km; weekly counts in the intersection of two 124 km disks, for the pairs whose training weeks average at least three events in the intersection; and the largest single-day count within 124 km of each base, or within 90 km under daily area counts. Errors are averaged over bases or pairs, weeks and sampling runs. The rewritten share of a correction step is the fraction of its output events that do not appear unchanged in its input, computed per week; Section 5.3 reports the median over weeks, averaged over runs. Diversity is the mean distance between the five samples generated for the same week. For independent draws from the same true conditional law, expected generated-to-generated and generated-to-data distances coincide; we use this identity as a dispersion diagnostic. Averaged over test weeks, these distances are 0.832 and 0.857 under weekly coverage budgets, and 0.417 and 0.465 under daily coverage budgets; projected DSTPP samples have similar ratios at larger distances.

E.4 SAMPLING COST

Partition budgets. Table 4 gives the time per window on one shared RTX 3090 GPU with batch size one, and Figure 8 plots it against the number of events under Total budgets. Each setting uses eight test windows at evenly spaced ranks of the event count, each timed three times after two warm-up runs; the table reports the median over windows of the per-window medians. Our time covers drawing the allocation and all reverse steps, and that of BD + Repair covers generation and repair. The windows hold 19–67 events on the synthetic datasets, 50–158 on EARTHQUAKE and 48–271 on EMS.

Wildfire weeks. Sampling times were measured on ten test weeks with one CPU process on a shared server: 27.7 s summed over the ten weeks under weekly and 28.5 s under daily coverage budgets. The couplings of the reverse steps take 59% and 55% of this time, null moves 30% and 36%, refresh 6% and 5%, initialization 4% and 2%, and the network 0.9% and 0.8%.

Table 3: Wildfire coverage budgets, all settings (156 test weeks; five samples per week). Local repair and DSTPP without a budget violate the budget in every week; all other rows meet it in every week. Planning errors: 90 km disks / intersections / busiest day (within 124 km under weekly and 90 km under daily area counts). Diversity / distance: mean distance between samples for the same week, and mean distance to the actual week.

Tier / method	Distance	MMD	CMMD ($\times 10^2$)	Planning errors	Diversity / distance
Weekly counts, air-attack areas only					
Ours	0.883	0.076	1.94	2.99/2.98/1.59	–
DSTPP + global projection	1.035	0.187	–	4.49/3.87/1.80	–
DSTPP, no budget	1.116	0.233	–	11.44/10.10/4.40	–
Weekly counts, air-attack and helitack areas					
Ours	0.857	0.066	1.92	2.53/2.68/1.63	0.832 / 0.857
DSTPP + global projection	0.980	0.148	2.55	3.71/3.67/1.76	0.941 / 0.980
DSTPP + global projection, relative placement	0.977	0.156	2.57	3.71/3.71/1.77	–
Ours sampled under daily totals, then global projection	0.902	0.117	2.13	3.40/3.30/1.73	–
relative placement	0.900	0.117	2.18	3.47/3.27/1.69	–
DSTPP + local repair, 100 sweeps	1.034	0.253	2.81	3.87/3.90/1.79	–
DSTPP + local repair, 20 sweeps	1.038	0.247	–	3.87/3.92/1.85	–
DSTPP, no budget	1.116	0.233	–	11.44/10.10/4.40	–
Daily counts, air-attack and helitack areas					
Ours	0.465	0.090	1.89	2.35/2.41/0.84	0.417 / 0.465
DSTPP + global projection	0.656	0.149	2.39	3.26/3.42/1.05	0.594 / 0.656
DSTPP + global projection, relative placement	0.649	0.156	–	3.23/3.44/1.03	–
Ours sampled under daily totals, then global projection	0.570	0.128	2.15	3.08/3.13/1.01	–
relative placement	0.555	0.128	2.22	3.06/3.04/1.02	–
DSTPP + local repair, 100 sweeps	0.795	0.245	2.73	3.85/3.94/1.13	–
DSTPP + local repair, 20 sweeps	0.815	0.247	–	3.88/3.97/1.20	–
DSTPP, no budget	1.116	0.233	–	11.44/10.10/2.98	–

Table 4: Sampling time per window in seconds, ours / BD + Repair. †Timed in a separate session.

Dataset	Total	Temporal	Spatial	Joint
SYN-1	1.01 / 0.50	0.98 / 0.55	0.94 / 0.88	0.94 / 3.75
SYN-2	0.97 / 0.47	0.96 / 0.54	0.93 / 0.85	1.62 / 6.65
EARTHQUAKE	1.10 / 0.49	1.09 / 0.55	1.03 / 0.89	1.73 / 6.70
EMS	2.87 / 0.49	2.67 / 0.55	3.40 / 1.66†	3.75 / 7.05

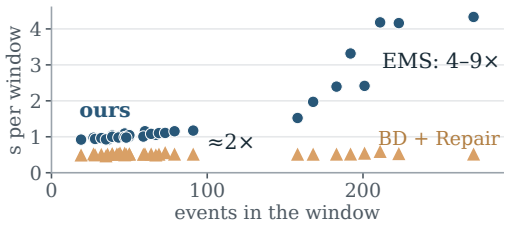


Figure 8: Sampling time per window under Total budgets.