LEARNING COLLECTIVE DYNAMICS WITH DIFFERENTIABLE GAUSSIAN REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Collective responses depend on individual differences, contact opportunities, and accumulated experience. Learning their dynamics from aggregate counts requires connecting a population's response distribution to both current observations and future behavior. We introduce **Differentiable Gaussian Dynamics (DGD)**, which learns this connection through three components: a Gaussian mixture representing heterogeneous response propensities, differentiable aggregation of contact intensity and behavioral probabilities, and feedback recurrence that updates subsequent responses. Reparameterized integration and temporal recurrence let aggregate prediction errors jointly train the distribution, observation functions, and feedback parameters. On four windows from KuaiRand-Pure and Online Retail II, DGD achieves lower joint behavioral negative log-likelihood than a DeepAR adaptation with a joint-behavior head. In Retail 2010, its one-day behavioral-count MAE is 4.71 versus 6.88 for this adaptation. Learning the distribution reduces behavioral negative log-likelihood by 10.82% relative to a fixed Gaussian in KuaiRand's standard-recommendation window; removing feedback dynamics raises joint KL from 0.0340 to 0.2577 in a controlled experiment. These results establish the value of learning population representations and their feedback process from aggregate observations.

1 INTRODUCTION

Social-media interactions and market transactions are collective outcomes of individual responses (Bass, 1969; Salganik et al., 2006). Their evolution depends on who participates, how frequently participants encounter an opportunity, and what they have experienced. Granovetter's account of collective behavior shows how distributions of individual thresholds determine group outcomes (Granovetter, 1978). For prediction, this raises a learning problem: how can aggregate event counts reveal the heterogeneous responses and feedback process that generate future collective behavior? Three coupled challenges determine the model this problem requires. **Response heterogeneity:** two audiences with the same mean propensity can have different response rates because behavioral probabilities are nonlinear; both differences between response types and variation within each type matter (Allenby & Rossi, 1998; Train, 2009). **Contact-dependent observations:** frequently reached participants contribute more events, so the response distribution in a log depends jointly on population composition and contact intensity (Liang et al., 2016; Schnabel et al., 2016; Gao et al., 2022). **Experience-dependent dynamics:** current contacts and responses accumulate as memory and fatigue (Kronrod & Huber, 2019; Kapoor et al., 2015), changing the distribution that will generate later observations (Chaney et al., 2018; Jiang et al., 2019). Aggregate counts combine these effects, so learning from them requires a path from the population state to current observations and from those observations to the next state.

Existing methods provide foundations for this task: mixture choice models integrate over preferences to predict aggregate choices (Train, 2009; Berry et al., 1995); aggregate dynamic models learn latent states and their evolution from population observations (Sheldon & Dietterich, 2011; Singh et al., 2022; Singh & Chen, 2023); world models predict subsequent observations through state transitions (Ha & Schmidhuber, 2018; Hafner et al., 2025); and differentiable simulation trains behavioral or transition parameters from observation errors (Mladenov et al., 2021; Chopra et al., 2023). Explicit Gaussian representations suggest a way to construct this path. In 3D Gaussian Splatting, differentiable image synthesis connects Gaussian scene parameters to observed pixels, allowing image errors to train

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.

the representation (Kerbl et al., 2023). We apply this representation–observation learning principle to collective responses: Gaussian components describe latent response propensities, and integration synthesizes observable population counts. As multiple image views constrain one scene, counts from different dates and observation units constrain a base response distribution shared across dates. Feedback then carries the consequences of current responses into the next prediction.

We introduce **Differentiable Gaussian Dynamics (DGD)**, with one component for each challenge. For response heterogeneity, a **Gaussian response representation** models type proportions, typical propensities, and variation within types through mixture weights, means, and covariances. For contact-dependent observations, **differentiable contact–behavior aggregation** evaluates contact intensity and behavioral probabilities at the same latent propensity, then integrates their product to predict arrivals (logged exposures or retail invoices) and behavioral totals. For experience-dependent dynamics, **feedback recurrence** uses observed or predicted outcomes to update memory, fatigue, response shifts, and the arrival environment. The components form one trainable model: integration propagates current prediction errors to the population representation, while recurrence lets later errors train the feedback process. During training, an expected-feedback branch also advances the state with predicted counts, as forecasting does.

We evaluate DGD on standard-recommendation and random-exposure windows from KuaiRand-Pure and two annual windows from Online Retail II. DGD has lower joint behavioral negative log-likelihood (NLL) than DeepAR-joint, a DeepAR adaptation with a joint-behavior head (Salinas et al., 2020), in all four windows, including 1.1232 versus 1.3466 under standard recommendation. Its one-day behavioral-count MAE is lower than DeepAR-joint and three historical-distribution baselines in both retail windows. Learning the distribution lowers standard-recommendation NLL from 1.2594 to 1.1232 relative to a fixed Gaussian, and feedback recurrence reduces random-exposure one-day behavioral-count MAE from 34.08 to 16.47. Controlled experiments recover known cross-day response distributions and predict choice distributions under new product attributes and prices.

Our contributions are:

- **A collective-dynamics model with an explicit population representation.** DGD connects a Gaussian response distribution, contact–behavior aggregation, and feedback recurrence to predict current and future collective responses.
- **Joint learning of representation and dynamics from aggregate counts.** Reparameterized integration and observed- and expected-feedback training propagate prediction losses to distribution, observation, and transition parameters within one objective.
- **Predictive evaluation and analysis of aggregate learning.** Four real-data windows and controlled experiments evaluate behavioral prediction, distribution learning, and feedback; theoretical and numerical analyses establish count-likelihood sufficiency, Gaussian parameter equivalence, and propagation of quadrature error through time.

2 Related Work

2.1 Learning Collective Responses from Population Observations

Threshold models connect individual differences to collective outcomes (Granovetter, 1978), while mixture choice models integrate conditional responses over continuous preference distributions or discrete latent classes (Train, 2009; McFadden & Train, 2000). Berry et al. (1995) estimate random-coefficient preference distributions from market shares, and Lu & Shimizu (2026) combine random-coefficient integration, market totals, and count likelihoods for demand estimation. These formulations motivate learning response distributions from population observations. Collective graphical models learn individual-level models from aggregate counts (Sheldon & Dietterich, 2011), and learning from aggregate observations trains instance-level predictors from group-level supervision (Zhang et al., 2020). Aggregate hidden Markov models learn latent dynamics through aggregate forward–backward inference and expectation–maximization (Singh et al., 2022; Singh & Chen, 2023), and population-dynamics models learn how distributions evolve from unpaired snapshots (Hashimoto et al., 2016; Tong et al., 2020; Bunne et al., 2022). Neural Hawkes models learn history-dependent event intensities (Mei & Eisner, 2017), and world models learn representations and transitions for predicting future observations (Ha & Schmidhuber, 2018; Hafner et al., 2025);

Appendix I reviews further event-intensity, count-forecasting, and simulation models. RecSim NG supports latent-variable learning in recommender ecosystems (Mladenov et al., 2021); GradABM calibrates behavioral mechanisms through gradients from aggregate outputs (Chopra et al., 2023), and differentiable agent-based simulation further develops gradient computation and aggregate supervision (Andelfinger, 2021; Quera-Bofarull et al., 2025). OranSim uses platform outcome predictions and population rollouts to select marketing scenarios (Ma et al., 2026). DGD uses a latent propensity shared by contact intensity and behavior and jointly learns its distribution, observation functions, and feedback transitions from daily count and behavioral-pattern losses.

2.2 Explicit Gaussian Representations

3D Gaussian Splatting optimizes explicit scene components through differentiable rendering (Kerbl et al., 2023). Dynamic Gaussian representations move or deform components over time to model changing scenes (Luiten et al., 2024; Wu et al., 2024). Gaussian world models predict scene states following robot actions (Lu et al., 2025), and ContactGaussian-WM learns physical parameters using Gaussian geometry, contact dynamics, and rendering (Wang et al., 2026). DGD's Gaussian components occupy response-propensity space. Differentiable integration maps this representation to population counts, and response feedback updates its subsequent state.

3 Differentiable Gaussian Dynamics

DGD has three components: a *Gaussian response representation* for heterogeneous propensities, *contact–behavior aggregation* for generating observations, and *feedback recurrence* for advancing the state (Figure 1). Sections 3.1–3.3 develop them in this order. Section 3.4 explains how aggregate losses train all three jointly.

Prediction task. An observation unit $j \in \{1, \dots, J\}$ contains disjoint cohorts $g \in \{1, \dots, G\}$. Units are creator streams in the social-media task and the whole store in the retail task. On date t, known features x_{jgt}, reference cohort size n_{jgt}, and nonnegative activity effort e_{jgt} accompany the observations. An arrival is a logged exposure in the social-media task and a retained merchandise invoice in the retail task; each arrival carries H binary behaviors $y \in \{0,1\}^H$. Daily pattern frequencies $C_{jgt}(y)$ give arrival counts $N_{jgt} = \sum_y C_{jgt}(y)$ and behavior counts $Y_{jgth} = \sum_y y_h C_{jgt}(y)$. From preceding history and known future inputs, DGD predicts expected arrivals $\widehat{N}_{jgt}$, joint probabilities $q_{jgt}(y)$ conditional on arrival, and expected behavior counts $\widehat{Y}_{jgth}$. The beginning-of-day state s_t is used to predict day t before that day's outcomes update s_{t+1}.

3.1 Gaussian Response Representation

We represent differences between and within response types with a K-component Gaussian mixture over latent propensity $U \in \mathbb{R}^d$:

$$P_{g,t}(u) = \sum_{k=1}^{K} \pi_{gk} \mathcal{N}(u; \mu_{gk} + \delta_{g,t}, \Sigma_{gk}), \quad \pi_g = \mathrm{softmax}(\omega_g), \quad \Sigma_{gk} = L_{gk} L_{gk}^\top, \tag{1}$$

where ω_g are mixture logits, π_{gk} is type k's population share, μ_{gk} its base mean, and Σ_{gk} its covariance. These parameters, denoted by θ_P, are learned and shared across dates. Feedback changes the cohort shift $\delta_{g,t}$, translating its components together. The lower-triangular factor L_{gk} has positive diagonals parameterized by softplus plus a positive floor (Pinheiro & Bates, 1996).

To make predictions differentiable with respect to the distribution, we transform fixed standard-normal integration nodes ϵ_i into component nodes by reparameterization (Kingma & Welling, 2014; Rezende et al., 2014):

$$u_{gkit} = \mu_{gk} + \delta_{g,t} + L_{gk} \epsilon_i. \tag{2}$$

Node locations move with the learned means and covariances, so errors in predictions evaluated at these nodes produce pathwise gradients for the representation (Mohamed et al., 2020). Each node has a positive weight w_i with $\sum_i w_i = 1$. We use Gauss–Hermite quadrature (Golub & Welsch, 1969) in one or two dimensions and fixed reparameterized Sobol nodes (Sobol', 1967; Buchholz et al., 2018) in higher dimensions.

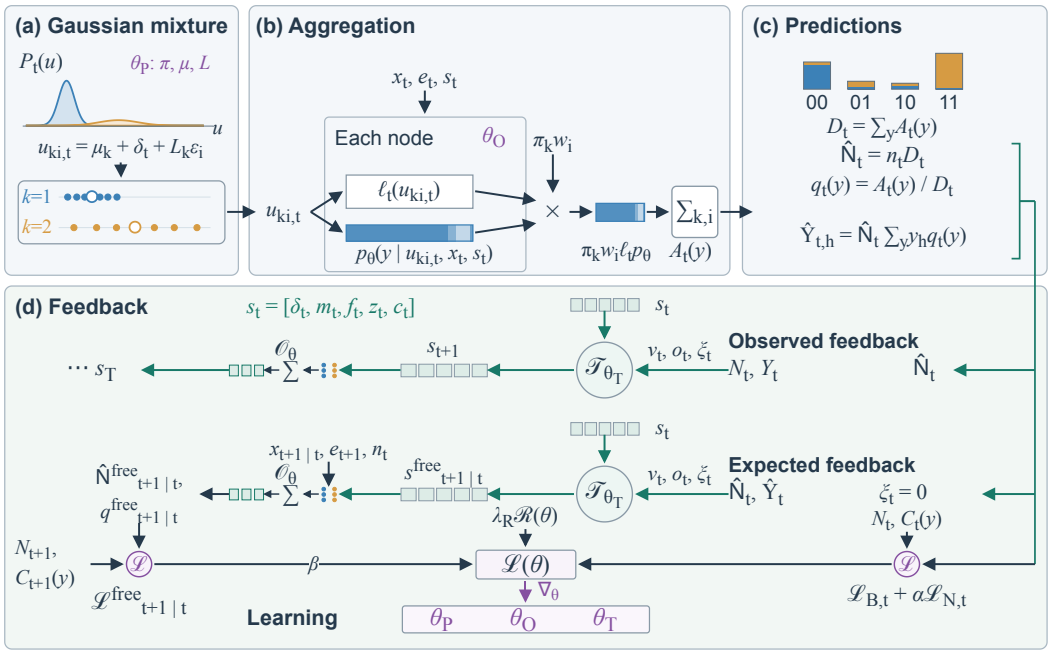


Figure 1: **Differentiable Gaussian Dynamics (DGD).** (a) The state translates a learned Gaussian mixture. (b) Arrival intensity and joint behavioral probabilities are multiplied at each node, then weighted and summed into pattern intensities $A_t(y)$. (c) Total intensity $D_t = \sum_y A_t(y)$ yields $\widehat{N}_t = n_t D_t$, $q_t(y) = A_t(y)/D_t$, and $\widehat{Y}_{t,h} = \widehat{N}_t \sum_y y_h q_t(y)$. (d) Observed and expected feedback advance the same initial state using recorded and predicted counts, respectively. Current and next-day expected-feedback losses train the population (θ_P), observation (θ_O), and transition (θ_T) parameters. Curves, nodes, and bars in (a)–(c) use a Retail-2010 fit, displaying two behaviors after marginalizing the third; unit and cohort indices are omitted.

3.2 Differentiable Contact–Behavior Aggregation

A type's contribution to recorded behavior depends on both its contact opportunities and its response after contact (Liang et al., 2016; Ma et al., 2018). DGD evaluates these quantities at the same latent propensity u. Per-person arrival intensity is

$$\ell_{jgt}(u) = e_{jgt} \exp\{\mathrm{clip}(a_{jg} + \beta_I^\top x_{jgt} + z_t + c_{jt} + \gamma^\top u, -16, 10)\}, \tag{3}$$

where a_{jg} combines baseline log-intensity and a learned intercept, β_I gives feature effects, z_t is the common arrival environment, and c_{jt} is the unit deviation. The learned arrival loading γ is shared across contexts. The operator clip restricts the log-intensity before the effort multiplier to $[-16, 10]$. Features are standardized using training-period statistics.

Conditional behavior follows an autoregressive logistic model (Neal, 1992; Larochelle & Murray, 2011):

$$\begin{aligned} p_\theta(y \mid u, x, s_t) &= \prod_{h=1}^{H} \sigma(\eta_h)^{y_h} [1 - \sigma(\eta_h)]^{1-y_h}, \\ \eta_h &= b_{jgh}(x, s_t) + \lambda_h^\top u + \sum_{h'<h} \Psi_{hh'} y_{h'}, \end{aligned} \tag{4}$$

where σ is the logistic function and η_h is behavior h's logit. The intercept function b_{jgh} contains unit and cohort intercepts, feature effects, and linear effects of memory, fatigue, and the common environment. The loading vector λ_h maps latent propensity to behavior h. We fix λ_1 to the unit vector along the first latent coordinate, anchoring its scale and direction (Bollen, 1989). The strictly lower-triangular matrix Ψ describes dependence on earlier entries in the same behavior vector. Shared propensity and within-arrival dependence both contribute to behavioral co-occurrence.

Let $\mathbb{E}_{P_{g,t}}$ denote expectation over the current propensity distribution. Integrating the contact–behavior product gives

$$\begin{aligned} \widehat{N}_{jgt} &= n_{jgt}\mathbb{E}_{P_{g,t}}[\ell_{jgt}(U)], \\ q_{jgt}(y) &= \frac{\mathbb{E}_{P_{g,t}}[\ell_{jgt}(U)p_\theta(y \mid U, x_{jgt}, s_t)]}{\mathbb{E}_{P_{g,t}}[\ell_{jgt}(U)]}, \\ \widehat{Y}_{jgth} &= \widehat{N}_{jgt}\sum_y y_h q_{jgt}(y). \end{aligned} \tag{5}$$

The numerator is the intensity of pattern y, and the denominator is the intensity of all arrivals. Their ratio is the probability that a recorded arrival exhibits pattern y. A small population component can therefore contribute many events if it is frequently reached (Patil & Rao, 1978). Multiplication before integration preserves the association between contact intensity and response probability.

At the reparameterized nodes, the behavioral probability is computed as

$$\widehat{q}_{jgt}(y) = \frac{\sum_{k,i}\pi_{gk}w_i\ell_{jgt}(u_{gkit})p_\theta(y \mid u_{gkit}, x_{jgt}, s_t)}{\sum_{k,i}\pi_{gk}w_i\ell_{jgt}(u_{gkit})}. \tag{6}$$

Section 3.4 uses this approximation and writes it as q. The same weighted sums produce arrivals and behavioral totals. We enumerate all 2^H patterns; positive weights preserve normalization. At zero effort, expected counts are zero and conditional probabilities use the positive-effort limit. The arrival and behavior parameters together form θ_O.

3.3 Feedback Recurrence

The state s_t contains response shifts $\delta_{g,t}$, behavioral memory $m_{g,t}$, fatigue $f_{g,t}$, and arrival states z_t, c_{jt}. Let $\widetilde{N}_{jgt}$ be the arrival count and $\widetilde{Y}_{jgt} = (\widetilde{Y}_{jgth})_{h=1}^H$ the vector of behavioral counts used for feedback. They contain recorded outcomes during observation and model expectations during forecasting. Cohort feedback comprises a smoothed behavioral-rate vector $v_{g,t}$ and contact dose $o_{g,t}$:

$$v_{g,t} = \frac{\sum_j \widetilde{Y}_{jgt} + 2v_g^0}{\sum_j \widetilde{N}_{jgt} + 2}, \qquad o_{g,t} = \frac{\sum_j \widetilde{N}_{jgt}}{\max(1, n_g^{\mathrm{ref}})}, \tag{7}$$

where v_g^0 is the reference behavioral-rate vector and n_g^{ref} is the maximum cohort reference size across units. Feedback updates the response state through

$$\begin{aligned} m_{g,t+1} &= \rho_m m_{g,t} + (1-\rho_m)v_{g,t}, \\ f_{g,t+1} &= \rho_f f_{g,t} + (1-\rho_f)(1-e^{-o_{g,t}}), \\ \delta_{g,t+1} &= \rho_\delta \delta_{g,t} + B^\top(v_{g,t} - v_g^0) - f_{g,t+1}b_f. \end{aligned} \tag{8}$$

The learned retention factors $\rho_m, \rho_f, \rho_\delta \in (0,1)$ control persistence through geometric carryover, a common form in marketing-response models (Hanssens et al., 2001; Jin et al., 2017); separate factors allow memory and fatigue to persist for different durations (Kronrod & Huber, 2019). The fatigue input $1 - e^{-o_{g,t}}$ saturates as contact dose grows, reflecting the diminishing marginal effect of repeated exposure (Krugman, 1972; Pechmann & Stewart, 1988). The matrix B maps behavioral-rate deviations to propensity shifts, and b_f maps fatigue to those shifts. Memory and fatigue also enter the behavioral logits directly. Current experience thus affects both the distribution and responses at a given propensity.

Arrival discrepancies update the common environment:

$$\xi_{jt} = \log(1 + \sum_g \widetilde{N}_{jgt}) - \log(1 + \sum_g \widehat{N}_{jgt}), \qquad z_{t+1} = \rho_z z_t + \kappa_z \bar{\xi}_t, \tag{9}$$

where ξ_{jt} is the unit's log-count innovation, $\bar{\xi}_t$ is its mean across active units, and ρ_z, κ_z are learned retention and correction coefficients. This correction follows the innovations form of exponential-smoothing state-space models (Hyndman et al., 2008). Unit states receive centered innovations and are centered to sum to zero (Appendix A). All transition parameters form θ_T. Forecasts fix historical summaries and reference sizes at the origin, advance known calendar and effort inputs, and feed predicted counts into the recurrence. Expected feedback gives zero arrival innovation while memory, fatigue, and response shifts continue evolving.

3.4 Joint Learning from Aggregate Counts

DGD learns $\theta = (\theta_P, \theta_O, \theta_T)$ through the complete prediction–feedback sequence. Arrival counts follow a negative-binomial working likelihood $\mathrm{NB}(N; \widehat{N}, r)$ with mean $\widehat{N}$, dispersion r, and variance $\widehat{N} + \widehat{N}^2/r$ (Cameron & Trivedi, 2013). Conditional on arrivals, pattern frequencies follow a multinomial likelihood with probabilities q. For daily cells $\mathcal{B}_t$, define

$$\mathcal{L}_{B,t} = -\frac{\sum_{(j,g)\in\mathcal{B}_t}\sum_y C_{jgt}(y)\log q_{jgt}(y)}{\max(1, \sum_{(j,g)\in\mathcal{B}_t} N_{jgt})}, \quad \mathcal{L}_{N,t} = -\frac{\sum_{(j,g)\in\mathcal{B}_t}\log \mathrm{NB}(N_{jgt}; \widehat{N}_{jgt}, r)}{|\mathcal{B}_t|}. \tag{10}$$

The parameter-independent multinomial constant is omitted. Behavioral losses are normalized by daily event counts and count losses by daily cell counts.

The observed branch advances the state with recorded counts, as in teacher forcing (Williams & Zipser, 1989), while forecasts advance it with predicted counts. To train the updates used in forecasting, an *expected-feedback branch* starts from the same s_t as the observed branch, uses predicted day-t counts to construct $s^{\mathrm{free}}_{t+1|t}$, and scores the next day's labels under the forecast input protocol in Section 3.3. Scheduled sampling and multi-step time-series training likewise feed a model's own predictions back during training (Bengio et al., 2015; Venkatraman et al., 2015). The branch loss is $\mathcal{L}^{\mathrm{free}}_{t+1|t} = \mathcal{L}^{\mathrm{free}}_{B,t+1|t} + \alpha\mathcal{L}^{\mathrm{free}}_{N,t+1|t}$. The objective over T training days is

$$\mathcal{L}(\theta) = \frac{1}{T}\left[\sum_{t=1}^{T}(\mathcal{L}_{B,t} + \alpha\mathcal{L}_{N,t}) + \beta\sum_{t=1}^{T-1}\mathcal{L}^{\mathrm{free}}_{t+1|t}\right] + \lambda_R\mathcal{R}(\theta), \tag{11}$$

where α, β, λ_R weight the count, forecast, and regularization terms. The parameter-normalized regularizer $\mathcal{R}$ penalizes main coefficients and departures of Gaussian scales from initialization. The computation graph is retained across the sequence, so gradients propagate backward through time (Werbos, 1990). Current losses train mixture weights and node locations through integration; later losses additionally train retention and feedback through recurrence. Algorithm 1 in Appendix G gives the complete procedure.

Properties of aggregate learning. For conditionally independent events sharing context and probability vector q, training from daily pattern counts yields the same parameter gradients as the event likelihood: their log-likelihoods differ only by a parameter-independent combinatorial term. Under unclipped exponential-linear intensity, arrival weighting shifts component means and reweights components, preserving a Gaussian mixture for recorded arrivals. With a shared arrival loading and common component shift, these changes can be absorbed into mixture parameters and the arrival intercept, yielding an equivalent separated readout. Expected feedback propagates finite-quadrature error to later states and predictions. Appendices B–C derive these properties and probability, gradient, and state-error bounds; Appendix F checks them numerically.

4 Experiments

We evaluate DGD's collective-response predictions, then examine how learning its population representation and feedback recurrence affects those predictions. Controlled experiments test response recovery under known dynamics and new product contexts.

4.1 Experimental Setup

Model instance. In all four real-data windows, DGD is a probabilistic model trained in PyTorch (Paszke et al., 2019) with 64-bit arithmetic. Each cohort has **two one-dimensional Gaussian components**, integrated with **seven Gauss–Hermite nodes per component**. The clipped exponential-linear arrival function and autoregressive logistic behavior function share the latent propensity. DGD enumerates 128 joint patterns for KuaiRand's seven behaviors and eight patterns for Retail's three basket marks. It learns distribution, observation, and transition parameters together; arrival counts use a negative-binomial likelihood with dispersion 50. Each window is fitted independently with three training seeds.

Data and prediction targets. KuaiRand-Pure provides standard-recommendation and random-exposure logs (Gao et al., 2022). Both windows use nine cohorts fixed from preceding history and five observation units: four creators and a pooled unit. Each exposure records clicking, long viewing, liking, following, commenting, forwarding, and negative feedback. Online Retail II provides two annual invoice windows (Chen, 2012). Each uses four customer cohorts and one store; the binary marks indicate merchandise value, quantity, and distinct-product count above warm-up medians. The three known-customer cohorts are single-purchase, low-spending repeat, and high-spending repeat customers; a fourth contains customers absent from warm-up or without identifiers. Features contain calendar inputs and preceding cohort summaries, and activity effort is $e = 1$. All windows use chronological train/validation/test splits (Table 8); preprocessing and dates are in Appendix D.

Baselines and fitting. DeepAR-joint adapts DeepAR's shared LSTM and negative-binomial count output with an autoregressive binary-mark head (Salinas et al., 2020). KuaiRand also includes two GRU predictors (Cho et al., 2014): a full-projection GRU learns every behavior's hidden-state projection, while a restricted-projection GRU fixes the first behavior's projection to the first hidden coordinate. Neural comparisons share the available aggregate inputs. Three historical baselines use training-period, last-day, or exponentially smoothed counts and joint distributions. DGD uses Adam (Kingma & Ba, 2015) with learning rate 0.025, gradient-norm clipping at 10 (Pascanu et al., 2013), and loss weights $\alpha = 0.05$, $\beta = 0.15$, $\lambda_R = 0.001$. Epoch budgets are 300 for KuaiRand and 120 for Retail, with patience 20. Validation selects epochs, DeepAR widths from 8 and 16, and historical smoothing settings. Appendix D supplies the complete protocols.

Metrics. Joint behavioral NLL averages the negative log-likelihood of each recorded event's joint behavior pattern over all test events. Mean Brier score averages squared errors between marginal probabilities and binary outcomes, equally across behaviors. Cohort-rate MAE weights cohorts equally and is reported in percentage points (pp). Behavioral-count MAE measures daily count errors averaged over behaviors or basket marks. Forecasts combine each model's own predicted arrivals and behavioral probabilities. Neural scores average three per-seed metrics; reported standard deviations describe training-seed variation. Appendix E gives scoring formulas and forecast settings.

4.2 Collective Response Prediction

DGD improves joint, marginal, and cohort-level behavioral prediction. Table 1 compares the four windows. DGD's joint NLL is lower than DeepAR-joint in every window and lower than both GRUs in the two KuaiRand windows. Under standard recommendation, NLL is 1.1232 versus 1.3466 for DeepAR-joint, while cohort-rate MAE is 4.21 versus 4.90 pp (Table 2). DGD also has lower Brier scores and cohort-rate errors than these recurrent comparators in each evaluated window. The improvements span the joint patterns used in training, individual behaviors, and equally weighted cohort responses. Section 4.3 examines the contribution of distribution learning.

Table 1: Joint behavioral NLL of DGD and recurrent baselines (lower is better). Values are means $\pm$ training-seed standard deviations; comparisons are within columns. DeepAR-joint adds a joint-mark head to DeepAR. GRUs were evaluated on KuaiRand; dashes denote unevaluated settings.

Model	Standard	Random	Retail 2010	Retail 2011
DGD	1.1232 ± 0.0005	0.7591 ± 0.0762	1.7357 ± 0.0012	1.7555 ± 0.0008
DeepAR-joint (adapted)	1.3466 ± 0.0224	0.7899 ± 0.0252	1.8819 ± 0.0205	1.8305 ± 0.0030
Full-projection GRU	1.2290 ± 0.0076	0.7970 ± 0.0094	—	—
Restricted-projection GRU	1.2307 ± 0.0202	0.7972 ± 0.0084	—	—

DGD also improves aggregate response-count prediction. In Retail, DGD has lower one-day behavioral-count MAE than every listed baseline in both annual windows (Table 3). Errors are 4.71 and 5.38 invoices/day, compared with DeepAR-joint's 6.88 and 5.77. The reductions relative to the lowest-error historical baselines are 54.60% and 54.48%, respectively. These predictions combine event volume and conditional behavior: the shared Gaussian representation generates arrivals and the probability that an invoice bears each basket mark. Appendix H plots the daily forecasts.

Table 2: Marginal and cohort-level predictions compared with recurrent baselines. Each cell gives mean Brier / cohort-rate MAE in percentage points (both lower is better). Neural scores average three training seeds. Appendix E defines the scores.

Model	Standard	Random	Retail 2010	Retail 2011
DGD	0.0683/4.21	0.0324/1.36	0.2482/12.84	0.2396/14.82
DeepAR-joint (adapted)	0.0709/4.90	0.0335/1.86	0.2535/14.26	0.2425/15.52
Full-projection GRU	0.0726/5.56	0.0325/1.39	—	—
Restricted-projection GRU	0.0731/5.62	0.0325/1.40	—	—

Table 3: One-day behavioral-count MAE in Retail (invoices/day; lower is better). Absolute daily errors are computed separately for high-value, high-quantity, and many-SKU marks, then averaged over dates and marks. Each window has 17 origins, with identical origin–endpoint pairs across methods. Relative improvements use unrounded errors.

Model	Retail 2010	Retail 2011
DGD	4.71	5.38
DeepAR-joint (adapted)	6.88	5.77
Training history	13.53	14.34
Last-day history	12.36	11.83
Smoothed history	10.38	12.29

4.3 Learning the Population Representation

The representation ablation retains the same observation functions, feedback structure, and aggregate histories. Fixed Gaussian freezes mixture weights, base means, and covariances; readout and transition parameters remain trainable. The matched discrete model uses three trainable support points, giving the same five effective distribution parameters per cohort as DGD; their initial means and total variances are matched. The point representation removes within-cohort heterogeneity.

Learning DGD's distribution reduces standard-recommendation NLL from 1.2594 to 1.1232, a **10.82% reduction** (Table 4). Brier, cohort-rate, and pairwise co-occurrence errors also decrease. Thus, gradients from aggregate observations improve the representation used to predict individual behaviors and their combinations. The point model has NLL 1.2443 and cohort MAE 5.74 pp. Gaussian and matched discrete mixtures both improve these predictions, attaining NLL 1.1232 and 1.1339, respectively.

Table 4: Representation and distribution-learning ablations under standard recommendation. Values are means $\pm$ training-seed standard deviations (lower is better). Rate errors are in percentage points (pp); co-occurrence error covers all 21 pairs of the seven behaviors.

Model	Joint NLL	Mean Brier	Cohort MAE (pp)	Pair MAE (pp)
DGD	1.1232 ± 0.0005	0.0683 ± 0.0001	4.21 ± 0.02	0.393 ± 0.016
Fixed Gaussian	1.2594 ± 0.0127	0.0761 ± 0.0006	6.36 ± 0.20	0.890 ± 0.115
Matched discrete	1.1339 ± 0.0074	0.0685 ± 0.0003	4.25 ± 0.10	0.404 ± 0.028
Point representation	1.2443 ± 0.0027	0.0734 ± 0.0002	5.74 ± 0.07	0.962 ± 0.020

4.4 Learning Feedback for Future Responses

The internal-dynamics ablation removes trainable recurrence while retaining observed lag summaries. Under random exposure, recurrence reduces one-day behavioral-count MAE from 34.08 to 16.47 and three-day MAE from 38.32 to 22.76 (Table 5). Because both variants retain the same observed summaries, the comparison evaluates the contribution of propagating current responses into subsequent state.

Table 5: Internal-dynamics ablation under random exposure. Values are means ± training-seed standard deviations of behavioral-count MAE, averaged over seven behaviors (occurrences/day). One-day and three-day forecasts have seven and five eligible origins, respectively.

Model	One-day MAE	Three-day MAE
DGD	16.47 ± 0.85	22.76 ± 0.50
Without internal dynamics	34.08 ± 0.59	38.32 ± 0.52

The controlled dynamics experiment fits the same arrival–response model to counts simulated from a generator with behavioral memory and response shifts. It uses two cohorts, two units, two behaviors, and two one-dimensional Gaussian components, with nine nodes per component and negative-binomial dispersion 80. The split contains 20 training, 8 validation, and 12 test days. DGD attains mean joint KL 0.0340 versus 0.2577 without dynamics (Table 6), with lower joint KL and marginal probability error in all three seeds. Sequence losses passing through recurrence recover the generator's cross-day response distributions from cohort counts.

Table 6: Response recovery under a known dynamic generator. KL compares predicted and true joint distributions; marginal MAE compares behavioral probabilities. Values are means ± seed standard deviations.

Model	Joint KL	Marginal MAE
DGD	0.0340 ± 0.0299	0.0519 ± 0.0337
Without dynamics	0.2577 ± 0.0052	0.1661 ± 0.0042

4.5 Response Prediction under New Attributes and Prices

This experiment evaluates responses to new product attributes and prices under known Gaussian and discrete generators. The controlled choice task independently fits a **two-component, two-dimensional Gaussian mixture** and a softmax readout over three products and an outside option. The readout models attribute preferences and positive price sensitivity; its integrated probabilities predict market-level purchase counts (Appendix A). Gaussian integration uses seven nodes per dimension. Each seed contains 20 training, 8 validation, and 40 test markets, each with 4,000 opportunities. The test markets are held out from fitting and validation.

Relative to a fixed Gaussian, learning the distribution reduces choice KL from 0.001994 to 0.001207 under the Gaussian generator and from 0.020016 to 0.002542 under the discrete generator (Table 7). The point representation has larger errors under both generators. These results show that mixture responses learned from aggregate purchases predict choice distributions in new product contexts.

Table 7: Choice-distribution recovery under known market generators. Values are the seed mean ± standard deviation of average KL across test markets.

Model	Gaussian generator	Discrete generator
Learned Gaussian	0.001207 ± 0.000544	0.002542 ± 0.002337
Fixed Gaussian	0.001994 ± 0.000522	0.020016 ± 0.027627
Point representation	0.059223 ± 0.005277	0.063599 ± 0.010622

5 Conclusion

DGD jointly learns a Gaussian response representation, contact–behavior aggregation, and feedback recurrence from aggregate counts. Across four recommendation and retail windows, it improves joint behavioral prediction over DeepAR-joint; ablations establish the contributions of distribution learning and feedback. Controlled experiments recover cross-day dynamics and responses under new attributes and prices.

Statement on the Use of AI Tools

AI tools assisted with manuscript organization, language editing, and preparation of the LaTeX source and diagrams. The reported experiments and numerical results derive from the research artifacts underlying this work. The authors are responsible for the content of the paper.

References

Greg M. Allenby and Peter E. Rossi. Marketing models of consumer heterogeneity. *Journal of Econometrics*, 89(1–2):57–78, 1998.

Philipp Andelfinger. Differentiable agent-based simulation for gradient-guided simulation-based optimization. In *Proceedings of the 2021 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, 2021. doi: 10.1145/3437959.3459261. URL https://philipp-andelfinger.net/pdfs/andelfinger2021differentiable.pdf.

Frank M. Bass. A new product growth for model consumer durables. *Management Science*, 15(5): 215–227, 1969. doi: 10.1287/mnsc.15.5.215.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Steven Berry, James Levinsohn, and Ariel Pakes. Automobile prices in market equilibrium. *Econometrica*, 63(4):841–890, 1995.

Kenneth A. Bollen. *Structural Equations with Latent Variables*. Wiley, 1989.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

Alexander Buchholz, Florian Wenzel, and Stephan Mandt. Quasi-Monte Carlo variational inference. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Charlotte Bunne, Laetitia Papaxanthos, Andreas Krause, and Marco Cuturi. Proximal optimal transport modeling of population dynamics. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, 2022.

A. Colin Cameron and Pravin K. Trivedi. *Regression Analysis of Count Data*. Cambridge University Press, 2nd edition, 2013.

Allison J. B. Chaney, Brandon M. Stewart, and Barbara E. Engelhardt. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018.

Daqing Chen. Online Retail II, 2012. URL https://archive.ics.uci.edu/dataset/502/online+retail+ii. Dataset, licensed under CC BY 4.0.

Daqing Chen, Sai Laing Sain, and Kun Guo. Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining. *Journal of Database Marketing & Customer Strategy Management*, 19(3):197–208, 2012.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1179.

Ayush Chopra, Alexander Rodríguez, Jayakumar Subramanian, Arnau Quera-Bofarull, Balaji Krishnamurthy, B. Aditya Prakash, and Ramesh Raskar. Differentiable agent-based epidemiology. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, pp. 1848–1857, 2023. URL https://aamas.csc.liv.ac.uk/Proceedings/aamas2023/pdfs/p1848.pdf.

Peter S. Fader, Bruce G. S. Hardie, and Ka Lok Lee. RFM and CLV: Using iso-value curves for customer base analysis. *Journal of Marketing Research*, 42(4):415–430, 2005.

René Ferland, Alain Latour, and Driss Oraichi. Integer-valued GARCH process. *Journal of Time Series Analysis*, 27(6):923–942, 2006. doi: 10.1111/j.1467-9892.2006.00496.x. URL https://doi.org/10.1111/j.1467-9892.2006.00496.x.

Konstantinos Fokianos, Anders Rahbek, and Dag Tjøstheim. Poisson autoregression. *Journal of the American Statistical Association*, 104(488):1430–1439, 2009.

Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. KuaiRand: An unbiased sequential recommendation dataset with randomly exposed videos. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, pp. 3953–3957, 2022. doi: 10.1145/3511808.3557624.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

Gene H. Golub and John H. Welsch. Calculation of Gauss quadrature rules. *Mathematics of Computation*, 23(106):221–230, 1969. doi: 10.1090/S0025-5718-69-99647-1.

Mark Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6): 1420–1443, 1978. doi: 10.1086/226707. URL https://doi.org/10.1086/226707.

David Ha and Jürgen Schmidhuber. World models, 2018. URL https://worldmodels.github.io/.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Bin Han and Bill Howe. SARN: Structurally-aware recurrent network for spatio-temporal disaggregation. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, 2024. doi: 10.1145/3678717.3691295.

Dominique M. Hanssens, Leonard J. Parsons, and Randall L. Schultz. *Market Response Models*. Springer, 2nd edition, 2001.

Tatsunori B. Hashimoto, David K. Gifford, and Tommi S. Jaakkola. Learning population-level diffusions with generative RNNs. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

Alan G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Rob J. Hyndman, Anne B. Koehler, J. Keith Ord, and Ralph D. Snyder. *Forecasting with Exponential Smoothing: The State Space Approach*. Springer, 2008.

Eugene Ie, Chih-wei Hsu, Martin Mladenov, Vihan Jain, Sanmit Narvekar, Jing Wang, Rui Wu, and Craig Boutilier. RecSim: A configurable simulation platform for recommender systems, 2019.

Ray Jiang, Silvia Chiappa, Tor Lattimore, András György, and Pushmeet Kohli. Degenerate feedback loops in recommender systems. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019.

Yuxue Jin, Yueqing Wang, Yuxiang Sun, David Chan, and Jim Koehler. Bayesian methods for media mix modeling with carryover and shape effects. Technical report, Google, 2017.

Komal Kapoor, Karthik Subbian, Jaideep Srivastava, and Paul Schrater. Just in time recommendations: Modeling the dynamics of boredom in activity streams. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, 2015.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian Splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023. doi: 10.1145/3592433. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Ann Kronrod and Joel Huber. Ad wearout wearout: How time can reverse the negative effect of frequent advertising repetition on brand preference. *International Journal of Research in Marketing*, 36(2), 2019.

Herbert E. Krugman. Why three exposures may be enough. *Journal of Advertising Research*, 12(6): 11–14, 1972.

Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011.

Dawen Liang, Laurent Charlin, James McInerney, and David M. Blei. Modeling user exposure in recommendation. In *Proceedings of the 25th International Conference on World Wide Web*, 2016.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. GWM: Towards scalable Gaussian world models for robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://gaussian-world-model.github.io/.

Zhentong Lu and Kenichi Shimizu. Estimating discrete choice demand models with sparse market–product shocks, 2026. URL https://arxiv.org/abs/2501.02381v3.

Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3D Gaussians: Tracking by persistent dynamic view synthesis. In *International Conference on 3D Vision*, 2024.

Jianxiang Ma, Mingfu Zhang, Xiaocui Yang, Yichen Gao, Junzhao Huang, and Yuesong Hou. OranSim: Simulating social media marketing, 2026. URL https://arxiv.org/abs/2609.28388v1.

Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2018.

Daniel McFadden and Kenneth Train. Mixed MNL models for discrete response. *Journal of Applied Econometrics*, 15(5):447–470, 2000.

Hongyuan Mei and Jason Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/6463c88460bd63bbe256e495c63aa40b-Abstract.html.

Martin Mladenov, Chih-Wei Hsu, Vihan Jain, Eugene Ie, Christopher Colby, Nicolas Mayoraz, Hubert Pham, Dustin Tran, Ivan Vendrov, and Craig Boutilier. RecSim NG: Toward principled uncertainty modeling for recommender ecosystems, 2021. URL https://arxiv.org/abs/2103.08057v1.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020.

Radford M. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 56(1):71–113, 1992.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

G. P. Patil and C. R. Rao. Weighted distributions and size-biased sampling with applications to wildlife populations and human families. *Biometrics*, 34(2):179–189, 1978.

Cornelia Pechmann and David W. Stewart. Advertising repetition: A critical review of wearin and wearout. *Current Issues and Research in Advertising*, 11(1–2):285–329, 1988.

José C. Pinheiro and Douglas M. Bates. Unconstrained parametrizations for variance-covariance matrices. *Statistics and Computing*, 6(3):289–296, 1996.

Arnau Quera-Bofarull, Nicholas Bishop, Joel Dyer, Daniel Jarne Ornia, Anisoara Calinescu, J. Doyne Farmer, and Michael Wooldridge. Automatic differentiation of agent-based models, 2025. URL https://arxiv.org/abs/2509.03303v2. Version 2, 18 November 2025.

Syama Sundar Rangapuram, Matthias W. Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Matthew J. Salganik, Peter Sheridan Dodds, and Duncan J. Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311(5762):854–856, 2006. doi: 10.1126/science.1121066.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020. doi: 10.1016/j.ijforecast.2019.07.001.

Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

Daniel R. Sheldon and Thomas G. Dietterich. Collective graphical models. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

Rahul Singh and Yongxin Chen. Learning gaussian hidden markov models from aggregate data. *IEEE Control Systems Letters*, 7:478–483, 2023. doi: 10.1109/LCSYS.2022.3187352. URL https://par.nsf.gov/servlets/purl/10397746.

Rahul Singh, Qinsheng Zhang, and Yongxin Chen. Learning hidden markov models from aggregate observations. *Automatica*, 137:110100, 2022. doi: 10.1016/j.automatica.2021.110100. URL https://arxiv.org/abs/2011.11236v2.

Ilya M. Sobol'. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967.

Henry Teicher. Identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 34(4): 1265–1269, 1963. doi: 10.1214/aoms/1177703862.

Alexander Tong, Jessie Huang, Guy Wolf, David van Dijk, and Smita Krishnaswamy. TrajectoryNet: A dynamic optimal transport network for modeling cellular dynamics. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Kenneth E. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, 2 edition, 2009. URL https://eml.berkeley.edu/books/choice2nd/Ch06_p134-150.pdf. Chapter 6: Mixed Logit, pages 134–150.

Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. Improving multi-step prediction of learned time series models. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Meizhong Wang, Wanxin Jin, Kun Cao, Lihua Xie, and Yiguang Hong. ContactGaussian-WM: Learning physics-grounded world model from videos, 2026. URL https://arxiv.org/abs/2602.11021.

Paul J. Werbos. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

Ronald J. Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4D Gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Sidney J. Yakowitz and John D. Spragins. On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 39(1):209–214, 1968.

Yivan Zhang, Nontawat Charoenphakdee, Zhenguo Wu, and Masashi Sugiyama. Learning from aggregate observations. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Kesen Zhao, Shuchang Liu, Qingpeng Cai, Xiangyu Zhao, Ziru Liu, Dong Zheng, Peng Jiang, and Kun Gai. KuaiSim: A comprehensive simulator for recommender systems. In *Advances in Neural Information Processing Systems*, volume 36, pp. 44880–44897, 2023. doi: 10.52202/075280-1945.

Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. SEISMIC: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

A Model Computation Details

The Gaussian mixture represents variation in latent response propensities within a cohort. The observation readout converts these propensities into response probabilities and event volumes under a given context. Social-media exposures and retail invoices use arrival and joint binary-behavior readouts. The controlled choice experiment pairs the same form of distributional representation with a choice-probability readout and fits it independently from market counts.

A.1 Aggregate Observation Likelihoods

The arrival likelihood describes the total number of events generated by a cohort within an observation unit; the behavioral likelihood describes the composition of their joint responses. Using the notation of the main text, the negative-binomial distribution has mean $\widehat{N}$, dispersion r, and log-probability

$$\begin{aligned}\log p(N \mid \widehat{N}, r) = \; & \log\Gamma(N+r) - \log\Gamma(N+1) - \log\Gamma(r) \\ & + r\log\frac{r}{r+\widehat{N}} + N\log\frac{\widehat{N}}{r+\widehat{N}}.\end{aligned} \tag{A.1}$$

Real-data training fixes $r = 50$ and sets the mean to the predicted arrival count. Given the arrival count N, behavioral-pattern counts C follow a multinomial distribution with joint probabilities q:

$$p(C \mid N, q) = \frac{N!}{\prod_y C(y)!}\prod_y q(y)^{C(y)}. \tag{A.2}$$

Within each date, the behavioral loss is divided by the number of events, with a minimum denominator of one, and the count loss is divided by the number of unit–cohort cells. The losses are then averaged over dates.

The expected-feedback forecast term starts from the current day's initial state, takes one expected-feedback step to obtain the next day's state, and evaluates behavioral and weighted count losses against the next day's aggregate labels. The validation objective contains only daily behavioral and weighted count losses. Test behavioral NLL is normalized by the total number of test events, using the event-weighted evaluation in the main text.

A.2 State Updates

The average prediction discrepancy across active units corrects the common arrival environment; each unit's departure from that average corrects its offset. Let $\mathbf{1}_{jt}$ indicate whether the sum, over cohorts, of cohort reference size times activity effort is positive for unit j on day t. Using the log innovations in Equation (9), the mean innovation across active units is

$$\bar{\xi}_t = \frac{\sum_j \mathbf{1}_{jt}\xi_{jt}}{\max(1, \sum_j \mathbf{1}_{jt})}. \tag{A.3}$$

Each unit's state retains part of its previous offset and is corrected according to the difference between its innovation and the mean innovation. The updated states are then centered across units:

$$\begin{aligned}\widetilde{c}_{j,t+1} &= \rho_z c_{jt} + \kappa_c \mathbf{1}_{jt}(\xi_{jt} - \bar{\xi}_t), \\ c_{j,t+1} &= \widetilde{c}_{j,t+1} - \frac{1}{J}\sum_{j'} \widetilde{c}_{j',t+1}.\end{aligned} \tag{A.4}$$

Centering keeps the sum of unit-specific states at zero, so these states represent offsets relative to the common arrival environment. The common environment follows Equation (9). Retention factors $\rho_\delta, \rho_z, \rho_f, \rho_m$ and correction gains κ_z, κ_c are all sigmoid-parameterized. Initial response shifts, memory, fatigue, and arrival states are zero; warm-up statistics set baseline intensities and reference behavioral rates.

During forecasting, model expectations provide feedback. With $\widetilde{N} = \widehat{N}$, the arrival innovation ξ_{jt} is zero, and common and unit-specific states decay with factor ρ_z. Behavioral memory, fatigue, and response shifts continue to follow Equations (7) and (8), advancing the cohort's response state to later dates.

A.3 Choice Response Readout

The controlled choice experiment learns heterogeneous consumer responses to product attributes and prices from market-level purchase counts, following random-coefficient logit demand models (Berry et al., 1995; McFadden & Train, 2000). Here, latent propensity u modulates attribute preferences and price sensitivity, and the choice readout converts it into probabilities of purchasing each product or making no purchase. Each product b has attributes x_b, price price_b, and an availability indicator. Conditional on u, its utility is

$$V_b(u) = x_b^\top \beta_C + u^\top W_C^\top x_b - \mathrm{price}_b \,\mathrm{softplus}(a_C + v_C^\top u), \tag{A.5}$$

where β_C gives baseline attribute preferences, W_C maps latent propensity to attribute preferences, and a_C, v_C determine the price-sensitivity intercept and loading. These are trainable choice-readout parameters. The softplus parameterization keeps price sensitivity positive. The outside option represents making no purchase on the current occasion and has utility zero. For the available product set $\mathcal{A}$, the softmax denominator includes only products in that set:

$$\begin{aligned} p(b \mid u) &= \frac{\exp V_b(u)}{1 + \sum_{b' \in \mathcal{A}} \exp V_{b'}(u)}, \quad b \in \mathcal{A}, \\ p(0 \mid u) &= \frac{1}{1 + \sum_{b' \in \mathcal{A}} \exp V_{b'}(u)}. \end{aligned} \tag{A.6}$$

Integrating choice probabilities over each cohort's propensity distribution, weighting by cohort reference size, and summing across cohorts gives aggregate demand $Q_b = \sum_g n_g \mathbb{E}_{P_g}[p(b \mid U)]$. Including the outside option yields $\sum_{b \in \mathcal{A} \cup \{0\}} Q_b = \sum_g n_g$, allocating all choice occasions to available products or no purchase. The choice experiment samples market counts from a multinomial distribution parameterized by aggregate probabilities and independently fits distribution and choice-readout parameters with the corresponding count objective.

B Equivalence under Aggregate Observations

B.1 Likelihood Sufficiency of Counts under Shared Conditions

Counts retain the parameter information in the event likelihood when the pooled events share the same conditional distribution. Suppose N events are conditionally independent given state and context and share joint probability $q(y)$. Collecting terms with the same behavioral pattern gives

$$\sum_{\nu=1}^{N} \log q(y_\nu) = \sum_y C(y) \log q(y). \tag{B.1}$$

There are $N!/\prod_y C(y)!$ event orderings with the same count vector, each with equal probability under the shared conditional distribution. Summing their probabilities adds a combinatorial constant to the log-likelihood above. This constant is independent of the parameters, so event and count objectives have identical parameter gradients. Collective graphical models likewise treat such count vectors as sufficient statistics of individual models for inference (Sheldon & Dietterich, 2011). Conditional independence applies across events; conditional dependence among behaviors within an event remains represented by $q(y)$.

B.2 Exponential Tilting of Gaussian Mixtures

Exponential arrival intensity changes the contribution of each response type to the observations, producing a weighted distribution of propensities (Patil & Rao, 1978). For the Gaussian mixture representing response heterogeneity, this reweighting can be expressed through a transformed set of mixture parameters. Fix a cohort and suppress its index. Let $\phi(u; m, \Sigma)$ denote the Gaussian density with mean m and covariance Σ. Completing the square in its exponent yields

$$e^{\gamma^\top u} \phi(u; m, \Sigma) = \exp\left(\gamma^\top m + \tfrac{1}{2}\gamma^\top \Sigma \gamma\right) \phi(u; m + \Sigma\gamma, \Sigma). \tag{B.2}$$

Substitute $m = \mu_k + \delta_t$ and define

$$\zeta_k = \exp\left(\gamma^\top \mu_k + \tfrac{1}{2}\gamma^\top \Sigma_k \gamma\right), \qquad Z = \sum_k \pi_k \zeta_k. \tag{B.3}$$

The cohort's dynamic shift contributes the same factor $\exp(\gamma^\top \delta_t)$ to every component, so this factor cancels after normalization. The original component weights are reweighted by ζ_k, and the component means shift in the direction determined by their covariances and the arrival loading. The propensity distribution reweighted by exponential intensity is therefore

$$\begin{aligned} P_E &= \sum_k \widetilde{\pi}_k \mathcal{N}(\widetilde{\mu}_k + \delta_t, \Sigma_k), \\ \widetilde{\pi}_k &= \frac{\pi_k \zeta_k}{Z}, \qquad \widetilde{\mu}_k = \mu_k + \Sigma_k \gamma. \end{aligned} \tag{B.4}$$

Set $\widetilde{P} = P_E$ as the response-propensity distribution of a separated-readout model. Directly integrating the same behavioral function gives $\widetilde{q}(y) = \mathbb{E}_{\widetilde{P}}[p(y \mid U)] = q(y)$. The observation weights assigned to response types by the original arrival intensity are now part of the transformed propensity distribution. Preserving arrival totals also requires an intercept adjustment. Define

$$\begin{aligned} \widetilde{Z} &= \sum_k \widetilde{\pi}_k \exp\!\left(\gamma^\top \widetilde{\mu}_k + \tfrac{1}{2}\gamma^\top \Sigma_k \gamma\right), \\ \widetilde{a} &= a + \log Z - \log \widetilde{Z}. \end{aligned} \tag{B.5}$$

The arrival part of the separated model retains the same arrival loading and uses the new intercept. The intercept adjustment cancels the change in integrated intensity caused by transforming the propensity distribution; the behavioral part uses the direct integral above. Both readouts therefore give the same arrival totals, joint behavioral probabilities, and behavioral totals. Identifiability results for finite Gaussian mixtures concern observed mixture densities (Teicher, 1963; Yakowitz & Spragins, 1968); this equivalence concerns integrals of the arrival and behavioral readouts.

If the models start from the same auxiliary states, deterministic observed or expected feedback maps their identical daily outputs to the same next-day states. Induction over days establishes path equivalence for any finite horizon.

This transformation applies to unclipped exponential intensity, an arrival loading shared across contexts, a dynamic shift shared by mixture components, and adjustable intercepts. The implemented quadrature model also includes intensity clipping and regularization. Section F evaluates this transformation numerically with fixed fitted parameters.

C Numerical Analysis of DGD

C.1 Probabilistic Properties of Positive-Weight Quadrature

Finite quadrature expresses the population response as a weighted sum of responses at latent nodes. Let $\widetilde{w}_{ki} = \pi_k w_i \ell(u_{ki}) \geq 0$ and assume $\sum_{ki} \widetilde{w}_{ki} > 0$. At each node, the autoregressive joint distribution sums to one across behavioral patterns. The normalized population probability therefore satisfies

$$\sum_y \widehat{q}(y) = \frac{\sum_{ki} \widetilde{w}_{ki} \sum_y p(y \mid u_{ki})}{\sum_{ki} \widetilde{w}_{ki}} = 1. \tag{C.1}$$

Expected behavioral counts are also nonnegative weighted sums of node-level expectations. Summing first within each Gaussian component and then across components only reorders a finite sum, so the component contributions sum to the population total. Each node's probability of a binary behavior lies between zero and one, giving $0 \leq \widehat{Y}_h \leq \widehat{N}$.

C.2 Quadrature Error Bounds

The joint behavioral probability is the ratio of behavior-weighted intensity to total arrival intensity, so quadrature error affects both its numerator and denominator. Let $q = A/D$ and $\widehat{q} = \widehat{A}/\widehat{D}$ be valid probabilities. Assume

$$D \geq d_0 > 0, \quad |\widehat{D} - D| \leq \varepsilon_D < d_0, \quad |\widehat{A} - A| \leq \varepsilon_A. \tag{C.2}$$

The identity

$$\widehat{q} - q = \frac{(\widehat{A} - A) - q(\widehat{D} - D)}{\widehat{D}} \tag{C.3}$$

together with $0 \le q \le 1$ bounds the numerator by the sum of the two integration errors and the denominator below by $d_0 - \varepsilon_D$, giving

$$|\widehat{q} - q| \le \frac{\varepsilon_A + \varepsilon_D}{d_0 - \varepsilon_D}. \tag{C.4}$$

To examine the effect on training gradients, further assume that derivatives of the integrals exist and satisfy

$$\begin{aligned} \|\nabla\widehat{A} - \nabla A\| \le \varepsilon_{\nabla A}, \quad \|\nabla\widehat{D} - \nabla D\| \le \varepsilon_{\nabla D}, \\ \|\nabla D\| \le G_D, \qquad\qquad \|\nabla q\| \le G_q. \end{aligned} \tag{C.5}$$

Write $\Delta_q = |\widehat{q} - q|$ and apply the quotient rule to the exact and approximate probabilities. Expand their gradient difference into errors in the numerator derivative, denominator derivative, probability value, and denominator value. The triangle inequality then gives

$$\|\nabla\widehat{q} - \nabla q\| \le \frac{\varepsilon_{\nabla A} + \varepsilon_{\nabla D} + \Delta_q G_D + G_q \varepsilon_D}{d_0 - \varepsilon_D}. \tag{C.6}$$

If $\min(q, \widehat{q}) \ge q_{\min} > 0$, then

$$\begin{aligned} |\log\widehat{q} - \log q| &\le \Delta_q / q_{\min}, \\ \|\nabla\log\widehat{q} - \nabla\log q\| &\le \frac{\|\nabla\widehat{q} - \nabla q\|}{q_{\min}} + \frac{G_q \Delta_q}{q_{\min}^2}. \end{aligned} \tag{C.7}$$

Weighting by empirical behavioral frequencies gives error bounds for average behavioral NLL and its gradient. Under reparameterized training, both mixture weights and latent node locations vary with the parameters, and their derivatives contribute to the integral's gradient.

C.3 Error Propagation and State Bounds under Expected Feedback

Expected feedback carries observation-integration errors into subsequent states. The recurrence error therefore includes both the approximation made in the current update and the propagation of existing state errors. Let exact and approximate expected-feedback updates be $s_{t+1} = \Phi_t(s_t)$ and $\widehat{s}_{t+1} = \widehat{\Phi}_t(\widehat{s}_t)$. Suppose Φ_t has Lipschitz constant L_t in the relevant trajectory region and $\sup_s \|\widehat{\Phi}_t(s) - \Phi_t(s)\| \le \varepsilon_t$. Then

$$\begin{aligned} E_{t+1} &\le L_t E_t + \varepsilon_t, \\ E_T &\le E_0 \prod_{t=0}^{T-1} L_t + \sum_{j=0}^{T-1} \varepsilon_j \prod_{t=j+1}^{T-1} L_t, \end{aligned} \tag{C.8}$$

where $E_t = \|\widehat{s}_t - s_t\|$. To obtain the one-step bound, add and subtract $\Phi_t(\widehat{s}_t)$ in the state difference. This separates the approximation error at the same input state, bounded by ε_t, from the exact update's response to the two input states, bounded by $L_t E_t$. Repeated substitution gives the finite-horizon sum. If $L_t \le L < 1$ and $\varepsilon_t \le \varepsilon$, then $E_T \le L^T E_0 + \varepsilon(1 - L^T)/(1 - L)$.

For fixed finite model parameters, suppose initial memory, fatigue, and reference behavioral rates lie in $[0, 1]$, and effort and cohort reference sizes are nonnegative. The behavioral-count bound in Appendix C.1 places smoothed behavioral rates under expected feedback in $[0, 1]$. Nonnegative dose also places $1 - e^{-o_{g,t}}$ in that interval. Memory and fatigue update by convex combinations and therefore remain in $[0, 1]$. Expected arrival innovations are zero, so common states and centered unit-specific states decay according to their retention factor.

Define $M_B = \max_l \sum_h |B_{hl}| + \|b_f\|_\infty$. Because the infinity norm of the difference between behavioral and reference rates is at most one, the response shift satisfies

$$\|\delta_t\|_\infty \le \rho_\delta^t \|\delta_0\|_\infty + \frac{M_B(1 - \rho_\delta^t)}{1 - \rho_\delta}. \tag{C.9}$$

The bound separates the decay of the initial shift from the accumulation of daily feedback along the deterministic expected-feedback path. The recurrence-error constants L_t depend jointly on retention factors, feedback coefficients, and the sensitivity of the observation readout.

D Experimental Protocols

D.1 Real-Data Preprocessing

Table 8: Real-data windows. Entries separated by slashes give training, validation, and test statistics. Events are exposures in KuaiRand and retained invoices in Retail.

Window	Days	Events	Behaviors
Standard	9 / 7 / 10	4,655 / 1,251 / 1,881	7
Random	4 / 3 / 7	2,256 / 2,033 / 9,101	7
Retail 2010	59 / 14 / 17	2,237 / 682 / 960	3
Retail 2011	59 / 14 / 17	2,159 / 565 / 869	3

Retail preprocessing removes duplicate rows, cancellations, nonpositive quantities or prices, non-merchandise entries, and conflicting records within invoices, retaining 39,434 positive-price merchandise invoices. Each annual window uses the preceding December for warm-up, January–February for training, March 1–14 for validation, and the remaining March dates for testing.

KuaiRand experiments retain recommendation-page logs (`tab=1`) from the candidate video pool. The standard-recommendation log records videos selected by the platform's recommendation policies. The random-exposure log records videos that KuaiRand inserted into recommendation lists with a fixed probability, each replacing one recommended video with a video sampled uniformly from the candidate pool (Gao et al., 2022). To assign users to buckets, we interpret the first eight bytes of the SHA-256 hash of each user identifier as a big-endian integer and take its remainder modulo 64. The main aggregate experiment uses bucket 1. The six previously fitted event-supervised models used for the analytic transformation in Section F come from bucket 0. The two samples contain disjoint users and use the corresponding calendar windows.

Event dates are recalculated from timestamps in the Asia/Shanghai time zone. Standard recommendation uses warm-up through April 12, 2022; training on April 13–21; validation on April 22–28; and testing on April 29–May 8. Random exposure uses warm-up through April 24; training on April 25–28; validation on April 29–May 1; and testing on May 2–8.

Cohort assignments are fixed before training. For standard recommendation, history through April 12 determines the cohorts and cohort reference sizes. For random exposure, these are determined from standard-recommendation history on April 8–21. Historical click rates are smoothed with 20 reference exposures and divided into four quantile groups, then crossed with two quantile groups of historical exposure volume to produce eight cohorts. New users enter a ninth cohort. The total reference sizes of the nine cohorts are 354 and 381 for standard recommendation and random exposure, respectively, including one smoothing unit per cohort. Observation units comprise the four creators with the most warm-up exposures and a pooled unit for all remaining creators.

The aggregate model's history summaries are updated only from previously released pattern counts; the event-level model additionally uses individual histories preceding exposures. The exponential moving averages of aggregate counts use retention 0.5 for KuaiRand and 0.85 for Retail. Compared models within each task share these summaries.

Retail 2010 and 2011 contain 1,665 and 1,549 warm-up invoices, respectively. The three known cohorts separate customers by purchase frequency and spending, as in RFM segmentation (Fader et al., 2005; Chen et al., 2012). Reference customer counts for these cohorts are 671, 140, and 140 in 2010 and 622, 131, and 131 in 2011. The cohort for customers absent from warm-up or missing customer identifiers uses one fixed reference unit. Warm-up medians of merchandise value, quantity, and distinct-product count are £294.52, 135, and 14 in 2010 and £254.40, 110, and 13 in 2011. Daily labels indicate values strictly above the corresponding thresholds. Dates with no transactions remain in the sequence with zero counts.

D.2 Model Evaluation Setup

The aggregate experiments use seeds 20260915, 20260916, and 20260917. The controlled choice and dynamics experiments use seeds 20260912, 20260913, and 20260914. DeepAR-joint is evaluated in

all four real-data windows, and the GRU predictors in the two KuaiRand windows. The full-projection GRU has 565 trainable parameters and is fitted under a separate supplementary protocol.

The main text reports each model's test results after validation-based epoch selection. The full-projection GRU uses a fully trainable hidden-state projection for every behavior. Its supplementary comparison was conducted after some test results for the original candidates were available, and its epochs were selected on validation data.

Behavioral-pattern NLL is conditional on a recorded exposure or retained invoice. Multistep behavioral evaluation uses cohort context frozen at the origin. Behavioral-count models and baselines are compared by MAE at identical forecast origins and endpoints. Longer horizons have fewer eligible origins within a fixed window; Tables 3 and 5 and Appendix E give the corresponding counts.

DeepAR implementation and selection. DeepAR-joint retains the shared LSTM (Hochreiter & Schmidhuber, 1997), count scaling, negative-binomial likelihood, and sampled autoregressive prediction of Salinas et al. (2020), and adds a joint-mark head. Our local implementation uses each short training sequence in full and weights cells uniformly in the count objective. Each cell is an observation-unit–cohort pair in KuaiRand or a cohort in Retail. Inputs concatenate the existing covariates, log-transformed cohort reference size and activity effort, a one-hot cell identifier, and the preceding count divided by $s_c = 1 + \mathrm{mean}_{\mathrm{train}}(N_c)$, where N_c is cell c's daily arrival count. Continuous input centers and scales are estimated on training data only. The initial previous count is zero, and subsequent training and validation steps use observed previous counts. A shared one-layer LSTM outputs NB2 mean $m_{\mathrm{NB}} = s_c \,\mathrm{softplus}(a)$ and dispersion $\alpha_{\mathrm{NB}} = \mathrm{softplus}(b)/\sqrt{s_c}$, adding 10^{-8} to each positive output before rescaling. The conditional variance is $m_{\mathrm{NB}} + \alpha_{\mathrm{NB}} m_{\mathrm{NB}}^2$.

The joint-mark head models $p(y_h \mid y_{<h}, \mathrm{state})$ with state-dependent logits and learned lower-triangular mark dependencies. Enumerating binary patterns gives a normalized joint distribution. Initial mark biases use training marginal rates. The training objective averages daily event-normalized mark NLL and adds 0.05 times mean cell count NLL and 10^{-3} times mean squared trainable parameters. Adam uses learning rate 0.01 and gradient-norm clipping at 10. Validation selects the epoch and hidden width from 8 and 16 under the unregularized objective. The 24 DeepAR-joint candidate fits (four windows, three seeds, and two widths) yield 12 selected models before test scoring. The supplementary protocol was specified after the original Gaussian and GRU test results were available; test predictions were produced after model selection, with no subsequent tuning.

DeepAR forecast protocol. At each test origin, the recurrent state contains only preceding observations. Non-calendar covariates, cohort reference size, and activity effort are frozen at their origin values; calendar coordinates advance with the date. Multistep predictions integrate 2,048 sampled negative-binomial count histories and average their conditional endpoint means. Conditional joint-mark probabilities average sampled-history probabilities weighted by their endpoint arrival means. After scoring an origin, the observed count enters the recurrent history for the next origin. One-day NLL weights each date's loss by its observed arrivals; behavioral-count MAE averages absolute daily errors over behaviors. Table means average three separately computed seed metrics. Supplementary results retain per-seed scores, daily predictions, input and checkpoint hashes, width and epoch selections, and forecast settings.

D.3 Numerical Comparisons

Representation comparisons match the number of trainable distribution parameters; numerical comparisons hold fitted parameters fixed and vary quadrature precision. The real-data Gaussian and matched discrete representations use two Gaussian components and three support points, respectively, with five effective distribution parameters per cohort. The KuaiRand numerical comparison includes four aggregate Gaussian variants (DGD, fixed Gaussian, independently trained separated readout, and no internal dynamics) and DGD trained on individual events. Retail includes DGD, fixed Gaussian, and no internal dynamics. Each configuration covers two windows and three training seeds. One-dimensional Gaussian training uses seven quadrature nodes. Numerical reevaluation holds fitted parameters fixed and uses 25 nodes; the analytic-equivalence evaluation additionally uses 61 nodes.

The independently fitted Gaussian representation in the controlled choice experiment has two dimensions and two components, with 11 effective distribution parameters per cohort. Gaussian

integration uses seven nodes per dimension. Choice models use Adam at learning rate 0.03, a parameter-normalized quadratic penalty weighted by 10^{-4}, a training budget of 180 epochs, and early-stopping patience 30. Only distribution and choice-readout parameters enter this optimization. The controlled dynamics experiment uses a one-dimensional, two-component Gaussian mixture, nine quadrature nodes, dispersion 80, a training budget of 150 epochs, and early-stopping patience 25.

E Behavioral Scoring and Forecast Settings

This appendix defines the historical baselines and behavioral scores used in the main text. Neural models use the three training seeds described there, and historical baselines are deterministic. Comparisons share data windows and forecast origins.

E.1 Historical Joint Distributions

Write $c = (j, g)$ for an observation-unit–cohort cell, and let $C_{ct}(y)$ be its count of behavioral pattern y on date t, with H behavioral dimensions. Training counts define a smoothed prior:

$$\bar{q}_y = \frac{\sum_{c,t\in\text{train}} C_{ct}(y) + 0.5}{\sum_{c,t\in\text{train},y'} C_{ct}(y') + 0.5\,2^H}, \qquad q_{ct}(y) = \frac{S_{ct}(y) + \tau\bar{q}_y}{\sum_{y'} S_{ct}(y') + \tau}. \tag{E.1}$$

Training-period history uses the sum of training pattern counts as S and predicts arrivals with the mean daily training count. Last-day history uses the most recently observed day's pattern counts and total count. Smoothed history exponentially averages daily pattern counts (Hyndman et al., 2008) and predicts arrivals with their sum. Its retention factor is $2^{-1/t_{1/2}}$, where $t_{1/2}$ is the half-life; zero-arrival dates also enter the update. Multistep predictions hold the distribution and arrival forecast fixed at the origin.

Each method selects smoothing strength from $\tau \in \{1, 10, 100\}$ on validation data. Smoothed history also selects its half-life from $t_{1/2} \in \{1, 3, 7\}$ days. Selection minimizes event-weighted validation joint NLL. Table E1 gives the selected retail settings; test scores are computed after selection. Each test date is predicted before its observations enter the history for the next origin. Neural models retain their selected checkpoints, whose original validation objective combines behavioral loss with weighted count loss. The supplementary behavioral evaluation was conducted after the original test windows had been analyzed; the added historical baselines follow the validation-selection rule above.

Table E1: Validation-selected settings for historical joint distributions in Retail.

Window	Training τ	Last-day τ	Smoothed τ	Half-life (days)
Retail 2010	10	100	1	7
Retail 2011	100	100	1	7

E.2 Probabilities, Behavioral Rates, and Counts

Let $N_{ct} = \sum_y C_{ct}(y)$, $Y_{cth} = \sum_y C_{ct}(y) y_h$, and $p_{cth} = \sum_y q_{ct}(y) y_h$. Event-weighted joint NLL and the Brier score (Brier, 1950) for behavior h are

$$\begin{aligned} \text{NLL} &= -\frac{\sum_{c,t,y} C_{ct}(y) \log q_{ct}(y)}{\sum_{c,t} N_{ct}}, \\ \text{Brier}_h &= \frac{\sum_{c,t} \left[Y_{cth}(1 - p_{cth})^2 + (N_{ct} - Y_{cth}) p_{cth}^2 \right]}{\sum_{c,t} N_{ct}}. \end{aligned} \tag{E.2}$$

Both are strictly proper scoring rules (Gneiting & Raftery, 2007). Mean Brier weights the H behaviors equally. Summing over cells gives observed behavioral counts, full predicted counts, and conditional predictions evaluated at observed arrival volumes:

$$Y_{th} = \sum_c Y_{cth}, \qquad \widehat{Y}_{th} = \sum_c \widehat{N}_{ct} p_{cth}, \qquad \widehat{Y}_{th}^{\text{cond}} = \sum_c N_{ct} p_{cth}. \tag{E.3}$$

Behavioral-count MAE is $H^{-1}\sum_h \mathrm{mean}_t\,|\widehat{Y}_{th} - Y_{th}|$. Conditional behavioral rates use $\widehat{Y}^{\mathrm{cond}}$ to evaluate response probabilities at the observed arrival composition. Observed arrival volumes enter conditional scoring; full count predictions use $\widehat{N}$.

Overall behavioral-rate MAE divides $100|\widehat{Y}^{\mathrm{cond}}_{th} - Y_{th}|$ by the total arrivals on that date, then averages over behaviors and dates with arrivals. Cohort-level scoring first pools observation units within each cohort, computes its mean date–behavior error in the same way, and then weights cohorts equally. Zero-arrival dates remain in count errors and are excluded from rate scores only when the corresponding rate denominator is zero.

E.3 Co-occurrence Rates and Forecast Horizons

For each behavioral pair (h, h'), the quantity $\sum_y q_{ct}(y) y_h y_{h'}$ predicts the probability that two behaviors occur within the same arrival. We weight these probabilities by observed arrival counts, compute daily co-occurrence-rate MAE, and average equally across all behavioral pairs. KuaiRand has 21 pairs and Retail has three.

Let Δ denote the forecast horizon. Multistep scoring evaluates the endpoint at day Δ, pairing each origin's prediction with the observation at the same endpoint. One-day scores use 10, 7, 17, and 17 origins in standard recommendation, random exposure, Retail 2010, and Retail 2011, respectively. The three-day random-exposure ablation in the main text uses five origins. Neural scores are computed separately for each training seed before their means are reported; seed standard deviations describe training variation within the same window.

F Numerical Evaluations of Aggregate Learning

Following the properties derived in Appendices B–C, numerical evaluations examine likelihood computation from counts under shared conditioning information, parameter transformations, and quadrature accuracy. We also check the numerical accuracy of gradients used in joint training and their propagation across days.

Under the shared conditioning information in Appendix B, the numerical difference between event and count objectives is zero after omitting the parameter-independent combinatorial constant, with a maximum gradient difference of 5.55×10^{-17}. In the gradient checks, central differences agree with automatic differentiation for mixture weights, means, and Cholesky parameters. Outputs across days have nonzero gradients with respect to retention, observation-correction, and feedback parameters.

The Gaussian tilting transformation in Appendix B maps an arrival-weighted population distribution to a corresponding separated readout. We apply this transformation to six sets of Gaussian model parameters trained on individual events, hold the fitted parameters fixed, and compare predictions before and after transformation. With 7, 25, and 61 quadrature nodes, the maximum absolute differences in mean test NLL are 9.65×10^{-4}, 1.01×10^{-5}, and 4.36×10^{-8}, respectively. As integration accuracy increases, the predictions of the two parameterizations converge, consistent with the analytic equivalence under exponential tilting of a Gaussian mixture.

Equation (C.4) describes the effect of integration error on predicted probabilities. We increase quadrature from the seven nodes used in training to 25 nodes, hold fitted parameters fixed, and recompute test behavioral losses. The KuaiRand comparison covers 30 fitted Gaussian models: four aggregate Gaussian variants and one individual-event training configuration, each with two windows and three seeds. The maximum absolute change in test NLL is 0.001747. For the standard-recommendation DGD models in Table 4, the maximum change is 0.000251. Retail includes three Gaussian variants across two windows and three seeds; the maximum change across these 18 models is 1.62×10^{-5}. Appendix D lists the variants and numerical settings.

G JOINT TRAINING AND DETAILED COMPUTATION

Algorithm 1 separates within-epoch state progression from parameter optimization. The observation operator $\mathcal{O}_\theta$ comprises Equations (1)–(6); the transition operator $\mathcal{T}_{\theta_T}$ consumes feedback counts and predicted arrivals to calculate state updates and arrival innovations. Figure 1 expands the node-level computation and the two feedback branches.

Algorithm 1 Joint learning of DGD from aggregate observations

Require: Training counts $C_{jgt}(y)$, known inputs, reference sizes, validation sequence.

Initialize $\theta_P, \theta_O, \theta_T$; compute feature scales from training data.
for each optimization epoch **do**
  Reset the daily state s_1 and accumulated loss $L \leftarrow 0$.
  for each training date $t = 1, \ldots, T$ **do**
    Predict $(\widehat{N}_t, q_t, \widehat{Y}_t) \leftarrow \mathcal{O}_\theta(s_t, x_t, n_t, e_t)$.
    Accumulate $L \leftarrow L + \mathcal{L}_{B,t} + \alpha \mathcal{L}_{N,t}$.
    if $t < T$ **then**
      Form $s^{\text{free}}_{t+1|t} \leftarrow \mathcal{T}_{\theta_T}(s_t, \widehat{N}_t, \widehat{Y}_t; \widehat{N}_t)$.
      Fix origin history summaries and sizes; advance known calendar and effort inputs.
      Predict from $s^{\text{free}}_{t+1|t}$ and score the next day's aggregate labels.
      Accumulate $L \leftarrow L + \beta \mathcal{L}^{\text{free}}_{t+1|t}$.
    end if
    Advance $s_{t+1} \leftarrow \mathcal{T}_{\theta_T}(s_t, N_t, Y_t; \widehat{N}_t)$ using observed feedback.
  end for
  Form $\mathcal{L} \leftarrow L/T + \lambda_R \mathcal{R}(\theta)$.
  Backpropagate through quadrature and the complete sequence; clip gradients.
  Update $\theta_P, \theta_O, \theta_T$ jointly using Adam.
  Select saved parameters and stopping epoch using the validation objective.
end for

H Daily Retail Behavioral-Count Forecasts

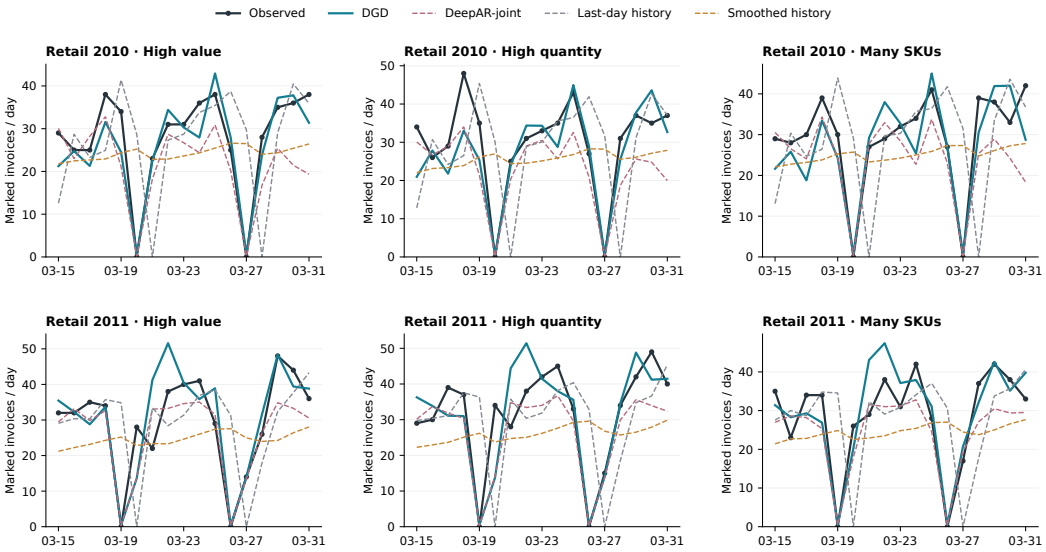


Figure 2: Daily behavioral-count forecasts in both retail windows. Columns show invoice counts for high value, high quantity, and many SKUs. Each forecast uses preceding observations and combines predicted arrivals with behavioral probabilities. Curves average predictions across three training seeds. Table 3 averages the errors computed separately for each seed.

I Additional Related Work

Hawkes processes model self-exciting event intensities and predict the popularity of social-media posts (Hawkes, 1971; Zhao et al., 2015). Probabilistic forecasters learn count distributions over time with recurrent or deep state-space models (Salinas et al., 2020; Rangapuram et al., 2018), and count autoregressions let past counts drive the conditional mean of later counts (Ferland et al., 2006; Fokianos et al., 2009). Recommender simulators generate user responses for evaluating recommendation policies (Ie et al., 2019; Zhao et al., 2023), and SARN disaggregates coarse spatio-temporal counts to finer levels (Han & Howe, 2024). DGD predicts such count sequences through a shared response distribution, contact-weighted aggregation, and feedback recurrence.