ALICE: IN-CONTEXT, ZERO-SHOT, MUTUAL INFORMATION ESTIMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Estimating mutual information (MI) from samples is a central objective in a variety of scientific fields. Modern neural estimators are accurate in the large-data regime, but they fall short when data is scarce, and each must be fit anew for every distribution under study. Current estimators are moreover tied to specific data types. These constraints limit their adoption in many applications where per-distribution training is impractical and sample sizes are small. We present ALICE, a foundation model that removes per-distribution training, while achieving competitive estimation accuracy. Trained exclusively on a broad family of synthetic distributions, ALICE acts as an in-context estimator of rectified-flow velocity fields: conditioned on samples of an unseen distribution, it estimates that distribution's velocity field without any explicit training. MI is then obtained through a fixed identity that integrates the squared difference between the joint and conditional fields. We validate ALICE on a standard, challenging benchmark and apply it in three domains, biology, genetics, and neuroscience, whose data the model has never seen. For the first time, we show that a single model closes the gap with neural estimators trained separately for each distribution, while natively supporting different data dimensionality and sample cardinality, enabling zero-shot MI analysis across scientific domains.

1 INTRODUCTION

Mutual Information (MI) quantifies the non-linear statistical dependence between two random variables (Shannon, 1948; MacKay, 2003) and is widely used in machine learning (Stratos, 2019; Belghazi et al., 2018; Oord et al., 2018; Hjelm et al., 2019), in biology (Nurse, 2008; Tostevin & Ten Wolde, 2009; Waltermann & Klipp, 2011; Brennan et al., 2012) and neuroscience (Borst & Theunissen, 1999; Ince et al., 2017; Nieh et al., 2021), to name a few. For random variables $X \in \mathbb{R}^{d_x}$ and $Y \in \mathbb{R}^{d_y}$, we write $Z = (X, Y) \in \mathbb{R}^d$ with $d = d_x + d_y$, and denote their joint distribution and marginals by p_{XY}, p_X, and p_Y. Their mutual information is the KL divergence

$$\mathrm{I}(X;Y) = \mathrm{KL}\left[p_{XY} \parallel p_X \otimes p_Y\right], \tag{1}$$

where $p_X \otimes p_Y$ is the product of the marginals, with density $p_X(x)\, p_Y(y)$. Estimating MI from finite samples is a difficult problem: the estimand depends on the full joint density, MI is unbounded and dominated by rare high-density events, and guarantees are fragile in high dimension (Paninski, 2003; Poole et al., 2019; McAllester & Stratos, 2020; Czyż et al., 2023).

Existing sample-based estimators share a structural limitation: each one is fit anew for every distribution. Variational bounds such as MINE (Belghazi et al., 2018), InfoNCE (Oord et al., 2018), NWJ (Nguyen et al., 2010), and SMILE (Song & Ermon, 2020), together with diffusion based estimators such as MINDE (Franzese et al., 2024), InfoBridge (Kholkin et al., 2026), FMMI (Butakov et al., 2026), require training from scratch for each distribution, with associated computational and tuning costs and a risk of failure. InfoAtlas (Hu et al., 2026) is an amortized alternative which uses a hypernetwork to sidestep per-distribution training, but it suffers from a non-negligible penalty in terms of accuracy. Appendix A discusses these estimators and other related work.

The diffusion-based estimators above build on score-based and flow-matching generative models, which represent a distribution by a time-indexed field attached to a noising process that maps clean samples to Gaussian noise (Song et al., 2021b; Lipman et al., 2022). In this work, we use the rectified-flow velocity as this field. Let p be a density on $\mathbb{R}^d$, let $Z_0 \sim p$ be a clean sample, let $\epsilon \sim \mathcal{N}(0,\, I)$

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.

be standard Gaussian noise independent of Z_0, and let $t \in [0, 1]$. The rectified-flow interpolant $Z_t = (1-t)Z_0 + t\epsilon$ connects data at $t = 0$ to noise at $t = 1$. The per-sample flow-matching target is the direction $Z_0 - \epsilon$, and the associated velocity field is its conditional mean at a noised point,

$$v_t(z) = \mathbb{E}_p[Z_0 - \epsilon \mid Z_t = z]\,. \tag{2}$$

The key link to MI is that, along a common rectified-flow path, the KL divergence between two distributions is a time integral of squared velocity differences (Guo et al., 2005; Franzese et al., 2024; Wang et al., 2026). For Equation (1), an equivalent form of this identity compares the joint velocity with the two block-conditional velocities, obtained by noising one block while holding the other clean, so MI estimation reduces to evaluating three velocity fields.

This formulation suggests an amortized estimator. We propose ALICE, a single Transformer network (Vaswani et al., 2017), trained once on synthetic distributions to predict their rectified-flow velocity fields from samples, in the spirit of amortized in-context predictors for tabular data (Hollmann et al., 2023) and function classes (Garg et al., 2022). At inference, a finite context of samples from an unseen joint distribution determines the field represented by the model, and a query specifies the point at which it is evaluated. Three masked queries provide the fields required to estimate MI with a fixed set of forward passes. The training corpus is entirely synthetic: a family of parametric distributions that is simple to define, cheap to sample, and easy to extend. ALICE generalizes to distributions and data types that the corpus does not contain (Section 4).

The absence of per-distribution training is key in the low-data regime: existing neural estimators achieve high accuracy only with hundreds of thousands of training samples per distribution and degrade sharply below that (Section 3), while datasets in biology or neuroscience, for example, often provide a few thousand pairs at most. In contrast, ALICE covers context sizes from only a few hundred samples to tens of thousands and is competitive across the whole range.

Our contributions are as follows. We present **ALICE** (Section 2), a foundation model for in-context estimation of velocity fields that can be used at any joint width and context length. ALICE is the first zero-shot MI estimator whose accuracy matches that of estimators trained per distribution. We **validate** ALICE (Section 3) on the "Beyond Normal" benchmark (Czyż et al., 2023). In the zero-shot setting, ALICE is competitive with trained neural estimators at their full budget and, with one thousand samples, is the most accurate estimator by a factor of at least two. We also report three scientific **applications** (Section 4) on data absent from the training corpus. In these applications, ALICE reproduces findings obtained with dedicated estimators and extends them, since the cost of a few forward passes per estimate allows analyses that per-distribution training makes impractical.

2 ALICE

ALICE is a MI estimator that amortizes velocity-field estimation across joint distributions. It is trained once, exclusively on a synthetic corpus of joint distributions, with a masked flow-matching objective. At inference, it conditions on samples from an unseen joint distribution and evaluates the joint and block-conditional velocity fields under three noising patterns; a fixed velocity identity combines their aligned block-wise differences into the estimate. Figure 1 summarizes these ideas.

This section develops the construction in four steps. We first derive the velocity-form identity and its Monte Carlo estimator in Section 2.1. We then define the context-conditioned velocity field in Section 2.2, describe the size-independent architecture in Section 2.3, and explain the synthetic training corpus and masked pretraining objective in Section 2.4.

2.1 MUTUAL INFORMATION ESTIMATION

MI is the KL divergence between the joint law p_{XY} and the product of its marginals $p_X \otimes p_Y$. For two densities following a common rectified-flow interpolant, this KL divergence is a time integral of squared differences between their velocity fields (Guo et al., 2005; Franzese et al., 2024; Wang et al., 2026; Butakov et al., 2026). We now derive the main velocity-form identity for MI, while we defer the full derivation and discussion to Appendix B.

Let $z_0 = (x_0, y_0) \sim p_{XY}$ be a clean joint sample, and let $\epsilon = (\epsilon_X, \epsilon_Y)$ be an independent standard Gaussian perturbation. The two components of z_0 define the X and Y blocks, each of which may

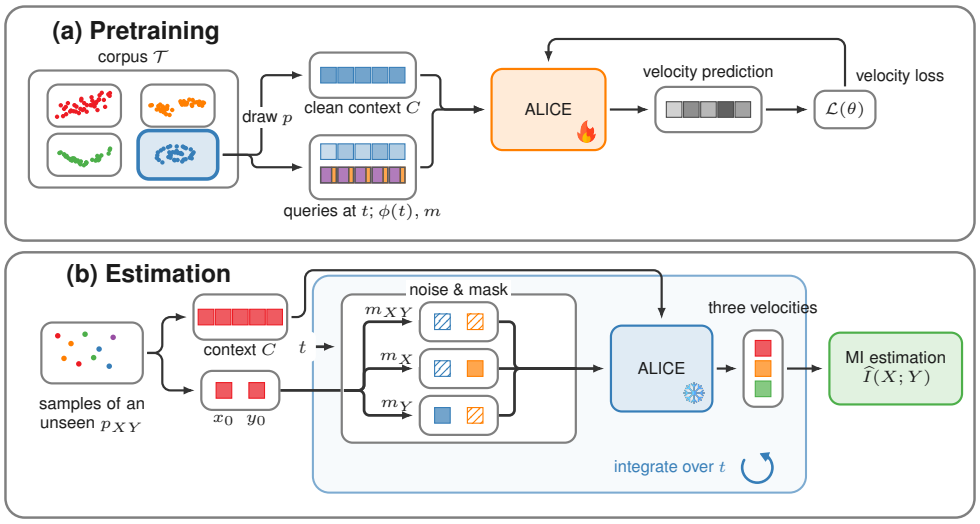


Figure 1: ALICE overview. **(a)** Pretraining: a distribution p drawn from the corpus $\mathcal{T}$ provides a clean context and queries noised according to indicator m; the model regresses the velocity target $z_0 - \epsilon$. **(b)** Estimation: samples of an unseen distribution provide a clean context and query $z_0 = (x_0, y_0)$; one shared Gaussian perturbation $\epsilon = (\epsilon_X, \epsilon_Y)$ at time t produces the three masked queries m_{XY}, m_X, and m_Y, whose block-wise velocity difference g, weighted by $(1-t)/t$, averages to the estimate.

contain multiple coordinates. For $t \in [0,1]$, diffuse the two blocks as $x_t = (1-t)x_0 + t\epsilon_X$ and $y_t = (1-t)y_0 + t\epsilon_Y$. Let $z_t = (x_t, y_t)$ denote the jointly noised point. For a concatenated vector $u = (u_X, u_Y)$, the selections $u|_X$ and $u|_Y$ retain the coordinates in the corresponding blocks.

Let $v_t(z)$ denote the joint velocity field. Let $v_t(x \mid y_0)$ denote the conditional velocity field of $p_{X|Y=y_0}$ evaluated at x. Let $v_t(y \mid x_0)$ denote the conditional velocity field of $p_{Y|X=x_0}$ evaluated at y. The resulting identity, proved in Appendix B (Theorem 2) and closest in mechanism to the decompositions of Franzese et al. (2024) and Wang et al. (2026), is:

$$\mathrm{I}(X;Y) = \int_0^1 \frac{1-t}{t}\, \mathbb{E}_{x_0,y_0,\epsilon}\Big[\, \|v_t(z_t)|_X - v_t(x_t \mid y_0)\|^2 + \|v_t(z_t)|_Y - v_t(y_t \mid x_0)\|^2 \,\Big]\, \mathrm{d}t. \quad (3)$$

In principle, Equation (3) involves three velocity fields: the joint field and two block-conditional fields. In practice, one can amortize these fields with a single model, represented by the parametric velocity field $v_\theta(z,t,m)$ for $z \in \mathbb{R}^d$ (Franzese et al., 2024). The mask $m \in \{0,1\}^d$ identifies the coordinates that are diffused and predicted and those held clean as evidence. The joint evaluation diffuses both blocks, and each conditional evaluation diffuses one block while holding the other clean.

We estimate the integral in Equation (3) with Monte Carlo. For each Monte Carlo draw i, sample a clean joint point $z_0^{(i)} = (x_0^{(i)}, y_0^{(i)})$, a time $t_i \sim \mathcal{U}[0,1]$, and one independent standard Gaussian perturbation $\epsilon^{(i)}$. Evaluating the definitions above at t_i gives the full query point $z_{t_i}^{(i)} = (x_{t_i}^{(i)}, y_{t_i}^{(i)})$. Define $m_{XY} = (\mathbb{1}_X, \mathbb{1}_Y)$, $m_X = (\mathbb{1}_X, \mathbb{0}_Y)$, and $m_Y = (\mathbb{0}_X, \mathbb{1}_Y)$, where $\mathbb{1}_X$ and $\mathbb{0}_X$ are the all-one and all-zero vectors on the X block, with the analogous convention for Y. Then, we have that

$$\widehat{v}^{(i)} = v_\theta(z_{t_i}^{(i)}, t_i, m_{XY}), \qquad \widehat{v}_X^{(i)} = v_\theta((x_{t_i}^{(i)}, y_0^{(i)}), t_i, m_X), \qquad \widehat{v}_Y^{(i)} = v_\theta((x_0^{(i)}, y_{t_i}^{(i)}), t_i, m_Y).$$

Using the same $\epsilon^{(i)}$ in all three model evaluations, the Monte Carlo estimator is

$$\widehat{\mathrm{I}}(X;Y) = \frac{1}{N_{\mathrm{MC}}} \sum_{i=1}^{N_{\mathrm{MC}}} \frac{1-t_i}{t_i} \left[\left\| (\widehat{v}^{(i)} - \widehat{v}_X^{(i)})|_X \right\|^2 + \left\| (\widehat{v}^{(i)} - \widehat{v}_Y^{(i)})|_Y \right\|^2 \right], \quad (4)$$

where N_{MC} is the number of samples (see Figure 1–b and Algorithm 1 in Appendix C for details).

Thus, our estimator requires a model that conditions on a clean sample context and accepts the query point, time, and noising indicator. The construct that meets these requirements is described next.

2.2 THE IN-CONTEXT VELOCITY FIELD

A conventional flow-matching model associates one set of parameters with one distribution p and approximates the map $(z_t, t) \mapsto v_t(z_t)$. Evaluating Equation (4) would require training one separate model for each distribution before its three velocity fields could be queried. We instead define, train, and use a single context-conditioned velocity model across a family of distributions. We represent this model as the map from a clean context *and* a query to a velocity, $(C, z_t, t, m) \mapsto v_\theta(z_t, t, m; C)$, where $C = \{z^{(k)}\}_{k=1}^{n}$ is a collection of n clean samples from the unseen distribution. At inference, the context determines *which* velocity field the in-context learning represents, while the query gives the argument *where* that field is evaluated. From a statistical learning perspective, the context size contributes to the *bias* of the estimator, while the query contributes to its *variance*. We implement this context-conditioned map with an attention-based transformer. A growing literature gives theoretical analyses and empirical demonstrations that transformers can implement learning procedures in their forward pass from in-context data (Garg et al., 2022; Akyürek et al., 2023; Von Oswald et al., 2023; Bai et al., 2023; Xie et al., 2022; Zhang et al., 2025; Xie et al., 2025). In a setting close to ours, Smart et al. (2025) show that a one-layer attention model can solve certain in-context denoising problems optimally. This motivates our approach, in which pretraining over a family of distributions teaches one shared attention model to infer the distribution-specific velocity computation from the context.

2.3 ARCHITECTURE

The architecture must process the context as a matrix $C \in \mathbb{R}^{n \times d}$, with one row per sample and one column per coordinate, for any context size n and joint width d with one set of parameters. Our method builds on recent work on in-context learning and set transformers, including the scalar tokenization of Chronos (Ansari et al., 2024), the any-variate attention of Moirai (Woo et al., 2024), and the amortized in-context inference of TabPFN (Hollmann et al., 2023) and TabICL (QU et al., 2025). The rows of the context are evidence about which distribution's velocity field to represent. The columns carry the dependence between coordinates, which the velocity of one coordinate needs from the values of the others. Appendix D complements the high-level description we discuss next.

Size-independent representation. Each scalar $z^{(k)}[i]$, coordinate i of context sample k, becomes one token, with one input projection and one scalar output head shared over all samples and coordinates. Context tokens contain a clean value and a type indicator; query tokens additionally contain the time features $\phi(t)$ and the noising-indicator entry $m[i]$. Since the context rows form a set and coordinate order is arbitrary, the model uses no positional encodings along either axis, and attention over the rows is applied separately to each column. A validity mask makes padded rows invisible, so the same parameters accept any context size n and are invariant to the order of the rows.

Induced latents. Self-attention over the n context tokens in every block would make compute and memory quadratic in n. To keep the cost linear in the context size, we introduce a bottleneck of K induced tokens that summarize the context for each coordinate, drawing inspiration from inducing variables in sparse Gaussian processes (Snelson & Ghahramani, 2005; Titsias, 2009), their use in deep Gaussian processes (Damianou & Lawrence, 2013; Salimbeni & Deisenroth, 2017), and induced set attention and latent-array architectures (Lee et al., 2019; Jaegle et al., 2021). The K induced tokens are the rows of one learned matrix $U \in \mathbb{R}^{K \times D}$, where D is the width of the token representations. For each coordinate, ALICE updates a copy of U through cross-attention over the context tokens, producing K context-specific latent vectors. For L blocks, this changes the context-dependent attention cost from $\mathcal{O}(Ldn^2)$ to $\mathcal{O}(dnK + LdK^2)$, which grows linearly in n for fixed K.

Context-derived relation graph. The velocity of one coordinate can depend on the values of other coordinates, and this dependence changes with the distribution. Separate marginal summaries cannot identify it: independently shuffling one context column preserves its marginal samples while changing which values occur together in a joint observation. We therefore construct a weighted graph with one node per coordinate, computed once from the clean context, whose edges control information exchange between coordinate representations; this follows the pattern of inferring interactions from observations to guide message passing (Kipf et al., 2018). The edge between coordinates i and j is derived from the covariance, over the context, of learned nonlinear features of $z^{(k)}[i]$ and $z^{(k)}[j]$, a principle also used in kernel dependence measures (Gretton et al., 2005). Nonlinear features expose relations such as $z[j] \approx z[i]^2$ that linear correlation misses, and centering the features makes the population descriptor vanish under independence. Each attention head turns this descriptor into a

signed, gated edge and uses the edges to mix the coordinate representations in every graph layer: within each context row before latent compression, and between latent and query representations.

Attention pattern. We use separate attention operations for context, latent, and query representations. Latent representations are updated by cross-attention from context tokens and by latent self-attention. Query representations attend to the latent and context representations and do not attend to one another, so each query is processed independently conditional on the same context-derived states. The shared attention and graph operations, together with the absence of positional encodings, preserve invariance to permutations of context samples and equivariance to permutations of coordinates.

Caching. The relation graph, the context tokens after the input projection, and the induced latents depend only on C, so they are computed once and reused across all queries for that distribution. Although this is not strictly useful during training, it is essential for inference on large contexts, where recomputing these states for every query would otherwise be costly.

2.4 PRETRAINING

We train ALICE to infer the velocity field of an unseen distribution from its context.

Pretraining corpus $\mathcal{T}$. Each episode is a synthetic joint distribution over $z = (x, y) \in \mathbb{R}^d$. The corpus combines base distributions (Gaussian, Student's t) with copula mixtures, latent warps, nonparametric regressions, and manifolds to vary dependence structure, conditional behavior, and support geometry. Copula mixtures follow the dependence-diversity construction of InfoAtlas (Hu et al., 2026) and use additive-coupling bijections (Dinh et al., 2017) to enrich the sampled dependencies. Latent warps transform Gaussian mixtures through shifts, folds, and rotations, nonparametric regressions generate responses from random Fourier-feature functions with noise, and manifolds generate near-singular supports. Appendix E.1 gives the full construction.

Training objective. The identity in Equation (3) reduces MI estimation to differences between a joint velocity field and masked conditional fields, so pretraining focuses on predicting these fields. The objective uses samples from each synthetic distribution and requires no MI labels, which lets one model amortize velocity-field estimation across the corpus $\mathcal{T}$. At each step, we sample a distribution $p \sim \mathcal{T}$, a context C of n independent samples from p, a further clean sample $z_0 \sim p$, a time t, Gaussian noise ϵ, and a noising indicator m. The indicator selects the coordinates that follow the interpolant in the query point $z_t = m \odot \big((1-t)z_0 + t\epsilon\big) + (1-m) \odot z_0$, while the remaining coordinates stay clean as evidence (Figure 1–a). We then regress the model output on the flow-matching direction over the selected coordinates:

$$\mathcal{L}(\theta) = \mathbb{E}_{p\sim\mathcal{T},\, C,\, z_0\sim p,\, t\sim\mathcal{U}[0,1],\, \epsilon,\, m} \left[\|m\|_1^{-1} \left\| m \odot \big(v_\theta(z_t, t, m; C) - (z_0 - \epsilon)\big) \right\|^2 \right]. \tag{5}$$

The factor $\|m\|_1^{-1}$ makes the loss a mean over noised coordinates, and the target contains no $1/t$ factor, so the training loss has no singularity as $t \to 0$. The noising indicator is sampled to cover the three fields required by Equation (3), all coordinates noised for the joint field and one block noised while the other remains clean for each block-conditional field, together with random coordinate subsets for general partial observation; since m is an input, one network represents all of these fields.

3 VALIDATION

We evaluate ALICE (Small and Base variants, see Appendix D) on the 40 tasks of the suite by Czyż et al. (2023), which spans joint widths from 2 to 100 and provides a closed-form ground-truth MI.

Protocol. Every estimator receives the same data budget of $N \in \{1\text{k}, 5\text{k}, 10\text{k}\}$ samples per task. ALICE splits the budget into 64 query samples and a context of the remaining $N - 64$ clean samples, and estimates MI in-context; estimates are in nats, averaged over eight independent context draws, and Appendix F gives the full inference settings. We compare against the following estimators: InfoAtlas (Hu et al., 2026), MINDE (Franzese et al., 2024), MINE (Belghazi et al., 2018), InfoNCE (Oord et al., 2018), D-V (Donsker & Varadhan, 1975), NWJ (Nguyen et al., 2010), KSG (Kraskov et al., 2004), LNN (Gao et al., 2015), and CCA (Hotelling, 1936). InfoAtlas is the amortized baseline and is evaluated zero-shot on the same contexts as ALICE. Neural estimators are trained (and tuned) separately on the N samples of each distribution, and classic estimators are fit directly on them.

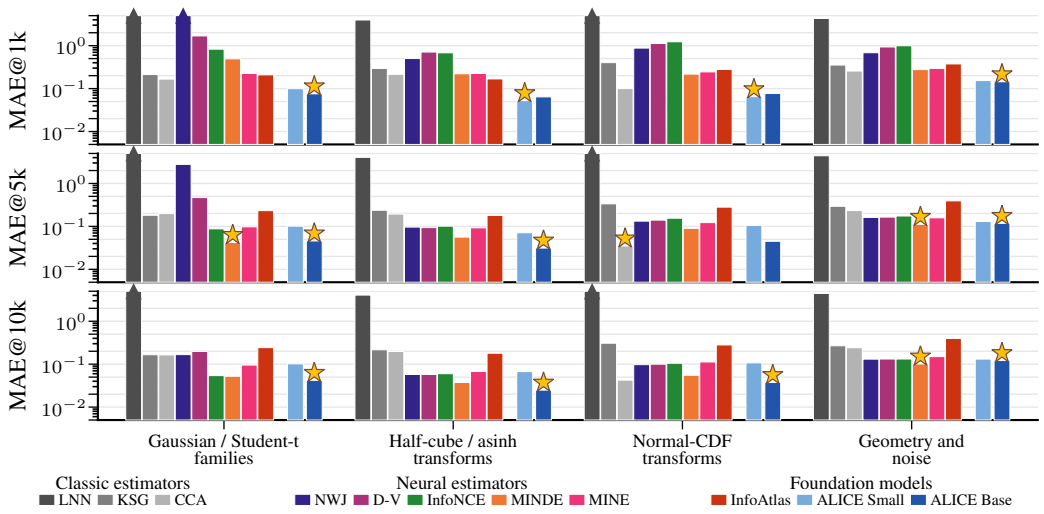


Figure 2: Category-wise MAE on the 40-task Czyż et al. (2023) benchmark at matched data budgets. Rows show budgets of 1k, 5k, and 10k samples, and columns group tasks by base family or transformation. Upward triangles mark values above the plotted range. Stars mark the lowest MAE.

Results. We report the mean absolute error (MAE) between estimate and ground truth over the 40 tasks, in nats; Figure 2 shows it per distribution group and Table 2 in Appendix F over the whole suite. ALICE Base has the lowest MAE at every budget: 0.092 nats at 1k samples, 0.063 at 5k, and 0.060 at 10k, against 0.195 for the best competitor at 1k (CCA) and 0.070 and 0.065 for MINDE at 5k and 10k. Within groups, ALICE has the lowest MAE in all four groups at 1k and in the three Gaussian-based groups at 10k. ALICE Small (19M parameters, against 85M for Base) has an MAE of 0.10 nats at every budget: second-lowest at 1k, and below MINE, D-V, NWJ, and the classic estimators at 5k and 10k. The two sizes differ on the wide tasks alone: on the 7 tasks of joint width 50 and 100 the MAE of Base falls from 0.153 nats at 1k to 0.086 at 10k while that of Small rises from 0.197 to 0.241; on the 33 narrower tasks they are within 0.02 nats of each other (Table 3). InfoAtlas has an MAE between 0.25 and 0.28 nats at every budget.

Small-data regime. The trained estimators need the step from 1k to 5k/10k samples to become practically usable: the MAE of MINDE falls from 0.353 to 0.070 nats, that of InfoNCE from 0.887 to 0.123, and that of D-V from 1.229 to 0.279, while NWJ diverges at 1k and is still at 1.261 nats at 5k. At a 1k budget, ALICE Base outperforms every competitor by a factor of at least two, and is more accurate than MINE, InfoNCE, D-V, and NWJ at 5k.

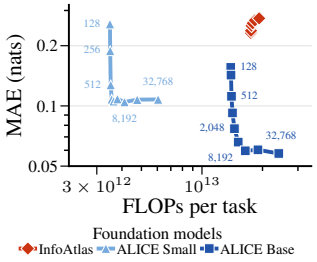
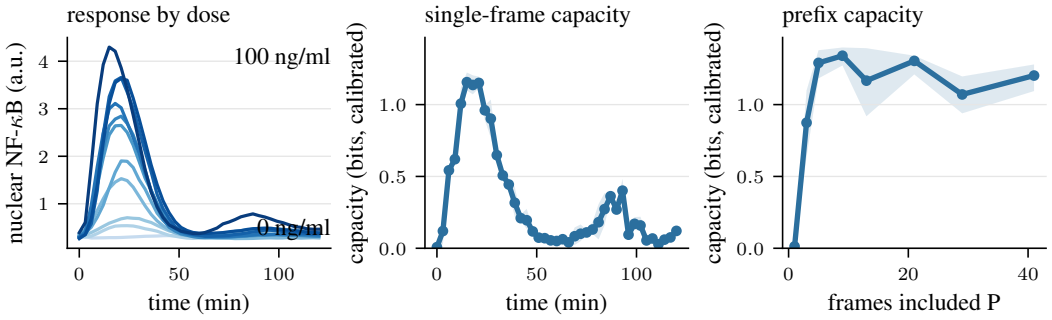


Figure 3: MAE against inference FLOPs per task and seed on the 40-task Czyż benchmark; labels show ALICE total budgets (64 queries).

Inference compute. Figure 3 compares operator-level FLOPs per task for ALICE and InfoAtlas. Here, ALICE uses a budget N from 128 to 32768, split into $N - 64$ context samples and 64 query samples, with 64 time draws per query; InfoAtlas uses contexts from 128 to 8192 samples. For each task, we average the MI estimates over 32 seeds for ALICE and eight seeds for InfoAtlas, then compute MAE across the 40 tasks. With a budget of 8192, Small and Base achieve MAEs of 0.105 and 0.060 nats for $4.1 \cdot 10^{12}$ and $1.6 \cdot 10^{13}$ FLOPs per task, respectively, compared with 0.274 nats for $1.9 \cdot 10^{13}$ FLOPs for InfoAtlas. Increasing the budget to 32768 gives Base an MAE of 0.058 nats for $2.4 \cdot 10^{13}$ FLOPs per task.

4 APPLICATIONS

In this section we showcase ALICE on three scientific applications: biology, genetics and neuroscience. In these applications, datasets include discrete distributions, sequences of tokens, and time-series of



(a) (b) (c)

Figure 4: In-context analysis of the NF-$\mathcal{K}$B dose channel. (a) Median nuclear NF-$\mathcal{K}$B response per TNF-α dose, with the interquartile band at the two extreme doses. (b) Capacity of each single frame and (c) of the prefix of frames from minute 0, with the envelope over seeds. The pairwise discrimination matrices are in Appendix G.

real numbers: not only ALICE's training corpus never encountered such distribution types, the model itself has never been trained on such data.

4.1 Analysis of multivariate single-cell signaling responses

Cellular signaling can be naturally described in information-theoretic terms: an extracellular stimulus X is transmitted through a stochastic biochemical network to a cellular response Y; $\mathrm{I}(X;Y)$ measures how reliably a cell can infer the stimulus, and the channel capacity is the maximum of $\mathrm{I}(X;Y)$ over input distributions (Nurse, 2008; Brennan et al., 2012; Jetka et al., 2018). We use ALICE to analyze the NF-$\mathcal{K}$B pathway, which responds to the inflammatory cytokine TNF-α (Jetka et al., 2019): 15,632 cells stimulated with one of $m = 11$ TNF-α concentrations (0 to 100 ng/ml) and imaged for 2 h at 3-min resolution (40 frames in total), the response being the nuclear-to-cytoplasmic NF-$\mathcal{K}$B ratio.

Protocol. MI decomposes as $\mathrm{I}(X;Y) = \sum_i p_i D_i$, where D_i is the divergence of the response distribution at dose i from the mixture over doses. We estimate every D_i from the velocity difference between a joint context and a label-shuffled context (see Appendix B.4 for a detailed formulation); ALICE returns both fields, with no training. From the D_i we obtain the MI at uniform input, the capacity by Blahut–Arimoto ascent, and the probability of correct discrimination (PCD) of every pair of doses, bracketed by the Jensen–Shannon divergence (see Appendix G for details). This is a small-data regime: after the filtering of the reference analysis, a dose has between 536 and 1,307 cell samples, and each D_i is estimated from a context of 1,024 cells.

Results. Figure 4 shows the three findings of the analysis. First, the information carried by a single frame follows the NF-$\mathcal{K}$B translocation: the capacity of one frame rises with the first nuclear peak, reaches about 1.2 bits at minutes 15 to 21, and decays, with a smaller second rise at the second peak (panel (b)). Second, the trajectory carries more than any single frame: the capacity of a prefix of frames saturates by frame 12 (panel (c)), so the dose is encoded in the timing and amplitude of the first response peak, and the prefix of frames reaches a capacity of about 1.3 bits against 1.2 for the best single frame. Third, dynamics separate the high doses: from a single frame, pairs of doses at or above 0.5 ng/ml are close to indistinguishable (mean PCD 0.56, where chance is 0.5), and the trajectory raises their PCD to 0.67, while the low doses are separable from a single frame already (Figure 10 in Appendix G). The three findings, the position and height of the capacity peak, and the discrimination pattern are obtained from a single model never trained on biological data, and not only corroborate those of Jetka et al. (2019), but overcome the limiting assumptions required approximate MI by fitting a linear classifier per analysis, which might not hold in more complex scenarios.

4.2 Promoter Identification

Regulatory motifs are short DNA patterns that control gene expression, and MI-based methods locate them by measuring the dependence between the content of a regulatory region and the expression it drives (Elemento et al., 2007; Rao et al., 2007). We use ALICE to locate the TATA-BOX, a core

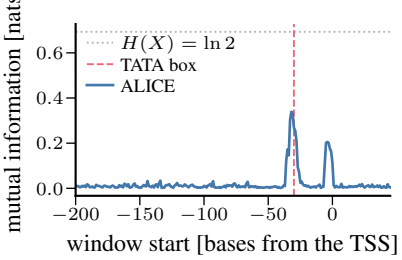


Figure 5: MI between the promoter label and a sliding window of $L = 6$, against the offset of the window start from the TSS; the dotted line marks the $\ln 2$ ceiling of the label entropy.

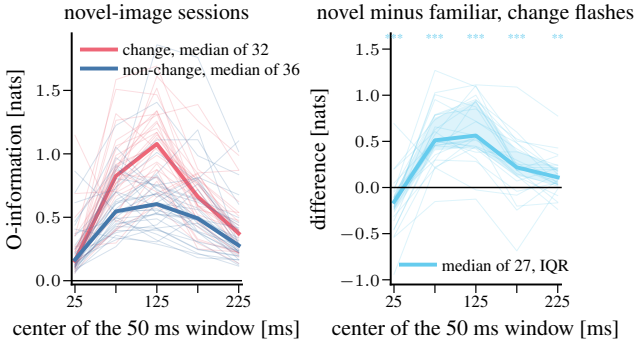


Figure 6: Ω-info of the six visual areas, one estimate per session. Left: novel-image sessions, one thin line per mouse and flash type, medians in bold. Right: novel minus familiar session for change flashes, with the median, the interquartile band, and a signed-rank test per window ($**$: $p < 0.01$, $***$: $p < 0.001$).

promoter motif whose preferred position in *Arabidopsis thaliana* lies 26 to 39 bases upstream of the transcription start site (TSS) (Bernard et al., 2010), on the promoter and non-promoter sequences of Umarov & Solovyev (2017) from the EPD database (Dreos et al., 2013): 1,497 sequences per class after balancing, each of 251 bases spanning positions -200 to $+50$ around the TSS, so the promoter label X is uniform and every MI value is bounded by $H(X) = \ln 2$ nats.

Protocol. For a window of $L \in \{4, 6\}$ bases starting at position s, we estimate $\mathrm{I}(X; Y)$ between the promoter label X and the window content Y. Sliding the window along the sequence produces a MI profile: windows on segments unrelated to promoter status yield near zero values, and windows overlapping an informative motif obtain high values. Each window is scored on its own, so a motif is detected even when another motif is correlated with it. Bases are input in the velocity fields of Equation (3) through a fixed injective embedding of one real coordinate per base, which preserves $\mathrm{I}(X; Y)$ exactly, and the blocks of dimension 1 and L are handled natively by ALICE. For every window position, ALICE conditions on a context of 1,024 label–window pairs and evaluates on 1,024 held-out pairs. This is a small-data regime: the whole dataset holds 2,994 sequences, an order of magnitude below the training sets that neural estimators require, and ALICE produces each window from 1,024 of them (see Appendix H for additional details).

Results. Figure 5 shows the profile for $L = 6$. The estimate is flat and near zero over the 200 bases upstream of the motif and over the 50 bases downstream of the TSS, rises sharply over the TATA-BOX band, with its maximum at 30 to 32 bases upstream of the TSS for both window lengths, and shows a second, smaller maximum on the TSS itself, which corresponds to the initiator element. The maximum lies inside the documented TATA-BOX band, which serves as a positive control for localization. The existing neural competitor for this task is INFO-SEDD (Foresti et al., 2026), a discrete-diffusion estimator trained on this dataset, which locates the TATA-BOX with windows realized by masking. Its profile scans positions -60 to -25 and reports a single peak, with a bias floor substantially higher than ALICE (see Appendix H for additional results).

4.3 BRAIN REGION ACTIVITY PATTERNS

We use ALICE to estimate the O-information (Ω-info) (Rosas et al., 2019) of six visual-cortex areas of mice performing a visual change-detection task, on the Visual Behavior Neuropixels recordings of the Allen Institute (Allen-Institute), first analyzed by Venkatesh et al. (2023). Bounoua et al. (2024) estimated the Ω-info of these recordings with a score-based estimator whose networks are trained on all sessions pooled together; we obtain the estimate from our frozen ALICE checkpoint, with no training on neural data, for every single session. For N random variables, the quantity $\Omega = \mathrm{TC} - \mathrm{DTC}$ is the difference between the total correlation and the dual total correlation; a positive value indicates that redundancy dominates the interactions, that is, the variables carry overlapping information, and a negative value indicates that synergy dominates. Both terms are time integrals of squared velocity differences of the form of Equation (4), between the joint field and the

concatenation of the N marginal fields (TC) or of the N conditional fields (DTC), so one checkpoint provides all the necessary fields (see Appendix I for details and validation).

Protocol. A mouse watches a natural image shown for 250 ms every 750 ms; the image repeats for several presentations (flashes) and then changes. We use the 72 sessions selected by Bounoua et al. (2024): 36 mice, each recorded on one day with a familiar image set and on another day with a novel one. For every flash, spikes are counted in five consecutive 50 ms windows and averaged over the units of each of six visual areas, so one flash is one draw of six variables and each window is one system of joint width six; change flashes and non-change flashes (repeats) are analyzed separately. Each session is estimated separately: ALICE conditions on a context of 128 flashes of the session and evaluates on the remaining flashes. Paired comparisons follow between the two flash types of a session and between the two sessions of a mouse. This is a small-data regime: a session provides about 150 to 200 independent flashes per flash type (Appendix I), too few to train an estimator per session, so Bounoua et al. (2024) pool all 72 sessions, and obtain no per-session estimate.

Results. Figure 6 shows how the six areas share information after a flash. In novel-image sessions the Ω-info is positive in every window and every mouse: the areas carry overlapping information. This redundancy is low at flash onset, maximal at 100 to 150 ms, when the visual response has reached all six areas, and decays afterwards. A change of image produces more redundancy than a repeat: in the peak window the within-session difference is positive in 31 of 32 sessions, and it is absent in the first window, before the visual response reaches the cortex. The same comparison in familiar-image sessions gives no difference at the peak and a reversed sign in the late windows, and the two sessions of each mouse (Figure 6, right) show that novelty raises the redundancy of the response in 25 of 27 animals for change flashes and in 28 of 36 for non-change flashes. We observe that a novel image drives a stimulus signal that is broadcast across the visual areas, and that this shared component fades with familiarity. The pooled result of Bounoua et al. (2024), a larger Ω-info after a change flash, therefore holds for novel images and in one window only, and the dependence on experience is a new finding of this work: the pooled analysis merges the two days of every mouse and cannot separate them. ALICE produced the 675 per-session systems in few forward passes of one frozen model; a trained-per-system estimator would require 675 training runs (Appendix I).

5 CONCLUSION AND LIMITATIONS

We presented ALICE, the first foundation model for MI estimation, whose zero-shot accuracy matches that of existing estimators trained per distribution. ALICE is a single Transformer, trained once as an in-context rectified-flow velocity field, that provides the joint and block-conditional fields of an unseen distribution from samples alone. A known identity uses such fields to estimate MI.

ALICE was trained exclusively on synthetic data and produced MI estimates zero-shot, on distributions and data types absent from its training corpus. On the "Beyond Normal" benchmark, it has the lowest error of all estimators at matched budgets of 1k, 5k, and 10k samples, although every competitor is trained and tuned on each distribution; at 1k samples, its error is lower by a factor of at least two. In three scientific applications, ALICE reproduces the findings of dedicated estimators in the small-data regime typical of biology and neuroscience, with a few forward passes per estimate.

We believe ALICE to be an invaluable asset for scientific discoveries across fields, that materializes as a local model that can be run "plug-and-play" on modest hardware.

Limitations. Our implementation is research code, and it has not been thoroughly optimized. Model size and training budget can be increased, training corpus can be augmented with higher dimensional data, maximum context size at training time can be increased, which might yield even better results in our benchmark validation.

ETHICS STATEMENT

MI estimates feed independence testing, feature selection, and scientific analysis, where a confidently wrong value can mislead downstream conclusions. We therefore advise caution on distributions that are far from the training corpus.

REPRODUCIBILITY STATEMENT

The code that runs the estimator of Algorithm 1 of this work is included in the supplementary material together with the checkpoint of ALICE Small. Appendices D and E specify the architecture, the corpus, the training procedure, and the hardware; Appendices C and F give the inference settings and the per-task benchmark results; Appendices G to I detail the data preparation and the protocol of each application.

AI USAGE STATEMENT

In this work, we used generative AI tools for help with LaTeX table formatting, TikZ figure polishing, and grammar and style polishing of the text; for generating figures and tables from stored results; for implementation design, polishing, and unit testing of the code; for low-level interaction with the GPU-cluster scheduler; and for managing artifacts, checkpoints, and datasets on the Hugging Face hub. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=0g0X4H8yN4I.

Allen-Institute. Visual behavior neuropixels dataset overview. URL https://brain-map.org/our-research/circuits-behavior/visual-behavior.

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=gerNCVqqtR.

Yaron E Antebi, Nagarajan Nandagopal, and Michael B Elowitz. An operational view of intercellular signaling pathways. *Current opinion in systems biology*, 1:16–24, 2017.

S. Arimoto. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 18(1):14–20, 1972.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. In *Advances on Neural Information Processing Systems (NeurIPS)*, 2023. URL https://openreview.net/forum?id=liMSqUuVg9.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R. Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning (ICML)*, 2018. arXiv:1801.04062.

Virginie Bernard, Véronique Brunaud, and Alain Lecharny. Tc-motifs at the tata-box expected position in plant genes: a novel class of motifs involved in the transcription regulation. *BMC genomics*, 11(1):166, 2010.

R. Blahut. Computation of channel capacity and rate-distortion functions. *IEEE Transactions on Information Theory*, 18(4):460–473, 1972.

Alexander Borst and Frédéric E Theunissen. Information theory and neural coding. *Nature neuroscience*, 2(11):947–957, 1999.

Mustapha Bounoua, Giulio Franzese, and Pietro Michiardi. Sωi: Score-based o-INFORMATION estimation. In *International Conference on Machine Learning (ICML)*, 2024. URL https://openreview.net/forum?id=LuhWZ2oJ5L.

Matthew D Brennan, Raymond Cheong, and Andre Levchenko. How information theory handles cell signaling and uncertainty. *Science*, 338(6105):334–335, 2012.

Ivan Butakov, Alexander Tolmachev, Sofia Malanchuk, Anna Neopryatnaya, and Alexey Frolov. Mutual information estimation via normalizing flows. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 3027–3057, 2024. URL https://openreview.net/forum?id=JiQXsLvDls.

Ivan Butakov, Alexander Semenenko, Valeriia Kirova, Ivan Oseledets, and Alexey Frolov. FMMI: Flow matching mutual information estimation. In *ICLR 2026 2nd Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2026. URL https://openreview.net/forum?id=2zTjX6rvn4.

Raymond Cheong, Alex Rhee, Chiaochun Joanne Wang, Ilya Nemenman, and Andre Levchenko. Information transduction capacity of noisy biochemical signaling networks. *Science*, 334(6054): 354–358, 2011.

Paweł Czyż, Frederic Grabowski, Julia E. Vogt, Niko Beerenwinkel, and Alexander Marx. Beyond normal: On the evaluation of mutual information estimators. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2306.11078.

Andreas Damianou and Neil D. Lawrence. Deep Gaussian processes. In Carlos M. Carvalho and Pradeep Ravikumar (eds.), *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pp. 207–215, Scottsdale, Arizona, USA, 2013. PMLR.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations (ICLR)*, 2017. arXiv:1605.08803.

M. D. Donsker and S. R. S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time, i. *Communications on Pure and Applied Mathematics*, 28(1):1–47, 1975.

Edo Dotan, Gal Jaschek, Tal Pupko, and Yonatan Belinkov. Effect of tokenization on transformers for biological sequences. *Bioinformatics*, 40(4):btae196, 2024.

René Dreos, Giovanna Ambrosini, Rouayda Cavin Périer, and Philipp Bucher. Epd and epdnew, high-quality promoter resources in the next-generation sequencing era. *Nucleic acids research*, 41 (D1):D157–D164, 2013.

Bell Raj Eapen. Genomic tokenizer: Toward a biology-driven tokenization in transformer models for dna sequences. *bioRxiv*, pp. 2025–04, 2025.

Olivier Elemento, Noam Slonim, and Saeed Tavazoie. A universal framework for regulatory element discovery across all genomes and data types. *Molecular cell*, 28(2):337–350, 2007.

Alberto Foresti, Giulio Franzese, and Pietro Michiardi. Information estimation with discrete diffusion. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=m18MXVdrV9.

Giulio Franzese, Mustapha Bounoua, and Pietro Michiardi. MINDE: Mutual information neural diffusion estimation. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2310.09031.

Shuyang Gao, Greg Ver Steeg, and Aram Galstyan. Efficient estimation of mutual information for strongly dependent variables. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015. arXiv:1411.2003.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://openreview.net/forum?id=flNZJ2eOet.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic Learning Theory*, volume 3734 of *Lecture Notes in Computer Science*, pp. 63–77. Springer, 2005. doi: 10.1007/11564089_7. URL https://www.cs.cmu.edu/~arthurg/papers/GreBouSmoSch05.pdf.

Dongning Guo, Shlomo Shamai, and Sergio Verdú. Mutual information and minimum mean-square error in Gaussian channels. *IEEE Transactions on Information Theory*, 51(4):1261–1282, 2005.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations (ICLR)*, 2019.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=cp5PvcI6w8_.

Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

Xixi Hu, Runlong Liao, Keyang Xu, Bo Liu, Yeqing Li, Eugene Ie, Hongliang Fei, and Qiang Liu. Improving rectified flow with boundary conditions. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 18177–18186. IEEE, 2025.

Zhengyang Hu, Yanzhi Chen, Hanxiang Ren, Qunsong Zeng, Youyi Zheng, Adrian Weller, Kaibin Huang, and Yanchao Yang. Infoatlas: A foundation model for zero-shot statistical dependence estimate. In *International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=VlspNGn7cK.

Robin A.A. Ince, Bruno L. Giordano, Christoph Kayser, Guillaume A. Rousselet, Joachim Gross, and Philippe G. Schyns. A statistical framework for neuroimaging data analysis based on mutual information estimated via a gaussian copula. *Human brain mapping*, 38(3):1541–1573, 2017.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning (ICML)*, pp. 4651–4664. PMLR, 2021.

Tomasz Jetka, Karol Nienałtowski, Sarah Filippi, Michael PH Stumpf, and Michał Komorowski. An information-theoretic framework for deciphering pleiotropic and noisy biochemical signaling. *Nature communications*, 9(1):4591, 2018.

Tomasz Jetka, Karol Nienałtowski, Tomasz Winarski, Sławomir Błoński, and Michał Komorowski. Information-theoretic analysis of multivariate single-cell signaling responses. *PLoS Computational Biology*, 15(7):e1007132, 2019.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. Scaling Laws for Neural Language Models. *ArXiv*, 2020.

Sergei Kholkin, Ivan Butakov, Evgeny Burnaev, Nikita Gushchin, and Alexander Korotin. InfoBridge: Mutual information estimation via bridge matching. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=y8Kzu9SKpv.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2688–2697. PMLR, 2018. URL https://proceedings.mlr.press/v80/kipf18a.html.

Xianghao Kong, Rob Brekelmans, and Greg Ver Steeg. Information-theoretic diffusion. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2302.03792.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6):066138, 2004.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. In *International Conference on Machine Learning (ICML)*, pp. 3744–3753. PMLR, May 2019. doi: 10.48550/arXiv.1810.00825.

Robin EC Lee, Sarah R Walker, Kate Savery, David A Frank, and Suzanne Gaudet. Fold change of nuclear nf-κb determines tnf-induced transcription in single cells. *Molecular Cell*, 53(6):867–879, 2014.

Maxwell W Libbrecht and William Stafford Noble. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6):321–332, 2015.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2022.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Aditya Malusare, Harish Kothandaraman, Dipesh Tamboli, Nadia A Lanman, and Vaneet Aggarwal. Understanding the natural language of dna using encoder–decoder foundation models with byte-level precision. *Bioinformatics Advances*, 4(1):vbae117, 2024.

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

Edward H Nieh, Manuel Schottdorf, Nicolas W Freeman, Ryan J Low, Sam Lewallen, Sue Ann Koay, Lucas Pinto, Jeffrey L Gauthier, Carlos D Brody, and David W Tank. Geometry of abstract learned knowledge in the hippocampus. *Nature*, 595(7865):80–84, 2021.

Paul Nurse. Life, logic and information. *Nature*, 454(7203):424–426, 2008.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *Advances in neural information processing systems (NeurIPS)*, 2018.

Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253, 2003.

Mariela D Petkova, Gašper Tkačik, William Bialek, Eric F Wieschaus, and Thomas Gregor. Optimal decoding of cellular identities in a genetic network. *Cell*, 176(4):844–855, 2019.

Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A. Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning (ICML)*, 2019. arXiv:1905.06922.

Jeremy E Purvis and Galit Lahav. Encoding and decoding cellular information through signaling dynamics. *Cell*, 152(5):945–956, 2013.

Lifeng Qiao, Peng Ye, Yuchen Ren, Weiqiang Bai, Chaoqi Liang, Xinzhu Ma, Nanqing Dong, and Wanli Ouyang. Model decides how to tokenize: Adaptive dna sequence tokenization with mxdna. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:66080–66107, 2024.

Jingang QU, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A Tabular Foundation Model for In-Context Learning on Large Data. In *International Conference on Machine Learning (ICML)*, 2025. URL https://openreview.net/forum?id=0VvD1PmNzM.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2008.02217.

Arvind Rao, Alfred O Hero III, David J States, and James Douglas Engel. Motif discovery in tissue-specific regulatory sequences using directed information. *EURASIP Journal on Bioinformatics and Systems Biology*, 2007:13853, 2007.

Fernando E. Rosas, Pedro A. M. Mediano, Michael Gastpar, and Henrik J. Jensen. Quantifying high-order interdependencies via multivariate extensions of the mutual information. *Physical review. E*, 100(3):032305, 2019. URL https://api.semanticscholar.org/CorpusID:67855406.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Jangir Selimkhanov, Brooks Taylor, Jason Yao, Anna Pilko, John Albeck, Alexander Hoffmann, Lev Tsimring, and Roy Wollman. Accurate information transmission through dynamic biochemical signaling networks. *Science*, 346(6215):1370–1373, 2014.

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3): 379–423, 1948.

Matthew Smart, Alberto Bietti, and Anirvan M. Sengupta. In-context denoising with one-layer transformers: Connections between attention and associative memory retrieval. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2502.05164.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. Platt (eds.), *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005.

Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. In *International Conference on Learning Representations (ICLR)*, 2020. arXiv:1910.06222.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021a. arXiv:2101.09258.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Karl Stratos. Mutual information maximization for simple and accurate part-of-speech induction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

Andrew E Teschendorff and Steve Horvath. Epigenetic ageing clocks: statistical methods and emerging computational challenges. *Nature Reviews Genetics*, 26(5):350–368, 2025.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In David van Dyk and Max Welling (eds.), *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pp. 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 2009. PMLR.

Filipe Tostevin and Pieter Rein Ten Wolde. Mutual information between input and output trajectories of biochemical networks. *Physical review letters*, 102(21):218101, 2009.

Ramzan Kh Umarov and Victor V Solovyev. Recognition of prokaryotic and eukaryotic promoters using convolutional deep learning neural networks. *PloS one*, 12(2):e0171410, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Praveen Venkatesh, Corbett Bennett, Sam Gale, Tamina K. Ramirez, Greggory Heller, Severine Durand, Shawn R Olsen, and Stefan Mihalas. Gaussian partial information decomposition: Bias correction and application to high-dimensional data. In *Neural Information Processing Systems (NeurIPS)*, 2023. URL https://openreview.net/forum?id=1PnSOKQKvq.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning (ICLR)*, pp. 35151–35174. PMLR, 2023.

Christian Waltermann and Edda Klipp. Information theory based approaches to cellular signaling. *Biochimica et Biophysica Acta (BBA)-General Subjects*, 1810(10):924–932, 2011.

Chao Wang, Luca Nepote, Giulio Franzese, and Pietro Michiardi. Relative entropy estimation in function space: Theory and applications to trajectory inference. In *International Conference on Machine Learning (ICML)*, 2026. URL https://openreview.net/forum?id=cpKJ2GlnYT.

Sean Whalen, Jacob Schreiber, William S Noble, and Katherine S Pollard. Navigating the pitfalls of applying machine learning in genomics. *Nature Reviews Genetics*, 23(3):169–181, 2022.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *International Conference on Machine Learning (ICML)*, 2024. URL https://openreview.net/forum?id=Yd8eHMY1wz.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Shifeng Xie, Rui Yuan, Simone Rossi, and Thomas Hannagan. The Initialization Determines Whether In-Context Learning Is Gradient Descent. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=fvqSKLDtJi.

Longxuan Yu, Xing Shi, Xianghao Kong, Tong Jia, and Greg Ver Steeg. MMG: Mutual information estimation via the MMSE gap in diffusion. In *Forty-Second Annual Conference on Uncertainty in Artificial Intelligence*, 2026. URL https://openreview.net/forum?id=qMHdwhu4kb.

Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. What and how does in-context learning learn? bayesian model averaging, parameterization, and generalization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2025. URL https://openreview.net/forum?id=B50OF0Fc6O.

A Related Work

We here expand on the closest prior works: per-distribution neural estimators, in-context inference with Transformers, and amortized estimation from synthetic corpora.

Variational MI estimation. Neural lower bounds (MINE (Belghazi et al., 2018), InfoNCE/CPC (Oord et al., 2018), NWJ (Nguyen et al., 2010), and the bias/variance study of SMILE (Song & Ermon, 2020) and Poole et al. (2019)) optimize a bound per distribution and are the standard against which diffusion estimators are measured. Classic kNN estimators (Kraskov et al., 2004) remain strong nonparametric baselines, and MIENF (Butakov et al., 2024) fits normalizing flows that separate the copula from the marginals.

Diffusion and information. MINDE (Franzese et al., 2024) expresses MI through a score-difference integral; information-theoretic diffusion (Kong et al., 2023) and the MMSE-gap estimator (Yu et al., 2026) develop the denoiser view; the velocity-form relative entropy of Wang et al. (2026) provides the basic estimator we use; InfoBridge (Kholkin et al., 2026) replaces score matching by bridge matching and obtains an exact drift-difference identity. All connect to the I-MMSE relation (Guo et al., 2005) and the likelihood weighting of Song et al. (2021a). These are the closest prior estimators based on *diffusion models*; each trains a network per distribution, which is the step ALICE amortizes.

Foundation models and in-context inference. Amortized in-context inference is realized by TabPFN (Hollmann et al., 2023) for tabular prediction, and that transformers learn function classes in-context is established broadly by Garg et al. (2022); ALICE adapts this inference mechanism to information estimation. Closest in the mechanism, Smart et al. (2025) study in-context denoising with one-layer transformers and its connection to associative memory (Ramsauer et al., 2021). The induced-latent context bottleneck follows set-attention and latent-array designs (Lee et al., 2019; Jaegle et al., 2021).

Amortized estimation and training corpora. The zero-shot claim depends on a broad synthetic training distribution. We extend the dependence-diversity design of InfoAtlas (Hu et al., 2026), random copula mixtures with coupling-flow (Dinh et al., 2017) augmentation, in the spirit of scaling-law-driven pretraining (Kaplan et al., 2020). InfoAtlas is the closest amortized estimator, and ALICE differs from it in three respects. First, InfoAtlas trains a hypernetwork that outputs the weights of a separate variational estimator for each distribution. ALICE keeps a single network and conditions it on the samples through attention, so no distribution-specific parameters are produced. Second, the coordinate-shared architecture of Section 2.3 is applied at any joint width, including widths absent from the corpus, while the weights a hypernetwork emits have a fixed shape and bind the estimator to the joint widths it was trained on. Third, ALICE estimates MI through the velocity-difference identity of Equation (3), which is exact for the true fields and involves no variational bound. Variational estimators output lower bounds, and a high-confidence lower bound above $\log N$ nats cannot be certified from N samples (McAllester & Stratos, 2020).

B Mutual Information as a Velocity-Difference Integral

This appendix proves Equation (3) in the notation of Sections 1 and 2.1. Theorem 1 expresses the KL divergence between two densities as a weighted time integral of the squared difference of their velocity fields, and Theorem 2 turns it into the joint-versus-conditional form the estimator uses. Results of the same kind exist in the I-MMSE relation of Franzese et al. (2024); Guo et al. (2005); Wang et al. (2026).

B.1 Velocity and score

As in Section 1, for a density p on $\mathbb{R}^d$, $Z_0 \sim p$, and $\epsilon \sim \mathcal{N}(0, I)$ independent of Z_0, the interpolant $Z_t = (1-t)Z_0 + t\epsilon$ has density p_t, and $v_t(z) = \mathbb{E}_p[Z_0 - \epsilon \mid Z_t = z]$ is the velocity field of p. When two densities are compared we mark the density as a superscript, v_t^p and v_t^q. Throughout, densities are assumed smooth with finite second moments and, for $t > 0$, with Gaussian tails, so that integrals can be differentiated under the sign and boundary terms of integrations by parts vanish; for $t > 0$ every p_t is a Gaussian convolution and has these properties.

Lemma 1 (Velocity and score). *For $t \in (0,1)$ and every z,*

$$\nabla \log p_t(z) = \frac{(1-t)\, v_t(z) - z}{t}. \tag{6}$$

Proof. Given Z_0, the noised point is Gaussian, $Z_t \sim \mathcal{N}\big((1-t)Z_0,\, t^2 I\big)$, so $p_t(z) = \mathbb{E}_p[\varphi_t(z - (1-t)Z_0)]$ with φ_t the density of $\mathcal{N}\big(0,\, t^2 I\big)$. Differentiating under the expectation and dividing by $p_t(z)$,

$$\nabla \log p_t(z) = -\frac{1}{t^2}\, \mathbb{E}_p[z - (1-t)Z_0 \mid Z_t = z] = -\frac{1}{t}\, \mathbb{E}_p[\epsilon \mid Z_t = z]\,,$$

since $z - (1-t)Z_0 = t\epsilon$ when $Z_t = z$. Taking conditional expectations in $z = (1-t)Z_0 + t\epsilon$ gives $z = (1-t)\mathbb{E}_p[Z_0 \mid Z_t = z] + t\,\mathbb{E}_p[\epsilon \mid Z_t = z]$; subtracting $(1-t)$ times the definition of $v_t(z)$ yields $\mathbb{E}_p[\epsilon \mid Z_t = z] = z - (1-t)v_t(z)$, and the claim follows. □

At $t = 0$ the interpolant is the clean sample and $\mathbb{E}_p[\epsilon \mid Z_0] = 0$ by independence, so $v_0(z) = z$ for every density: all velocity fields agree at the boundary.

Lemma 2 (Continuity equation). *For $t \in (0,1)$, $\partial_t p_t = \nabla\cdot(p_t\, v_t)$.*

Proof. Along each sample path $\frac{\mathrm{d}}{\mathrm{d}t} Z_t = \epsilon - Z_0$. For a smooth compactly supported test function ϕ, the tower property and the definition of v_t give

$$\frac{\mathrm{d}}{\mathrm{d}t}\, \mathbb{E}_p[\phi(Z_t)] = \mathbb{E}_p[\nabla\phi(Z_t)\cdot(\epsilon - Z_0)] = -\mathbb{E}_p[\nabla\phi(Z_t)\cdot v_t(Z_t)] = -\int \nabla\phi\cdot v_t\, p_t\, \mathrm{d}z.$$

The left side equals $\int \phi\, \partial_t p_t$, and integrating the right side by parts gives $\int \phi\, \nabla\cdot(p_t v_t)$. □

B.2 KL DIVERGENCE IN VELOCITY FORM

Theorem 1 (KL divergence as a velocity-difference integral). *For two densities p and q on $\mathbb{R}^d$ with $\mathrm{KL}\,[p \parallel q] < \infty$,*

$$\mathrm{KL}\,[p \parallel q] = \int_0^1 \frac{1-t}{t}\, \mathbb{E}_{z\sim p_t}\Big[\big\|v_t^p(z) - v_t^q(z)\big\|^2\Big]\, \mathrm{d}t. \tag{7}$$

Proof. Let $F(t) = \mathrm{KL}\,[p_t \parallel q_t] = \int p_t \log(p_t/q_t)$. At $t = 1$ both interpolants equal ϵ, so $p_1 = q_1 = \mathcal{N}(0,\, I)$ and $F(1) = 0$; at $t = 0$, $F(0) = \mathrm{KL}\,[p \parallel q]$. Hence $\mathrm{KL}\,[p \parallel q] = -\int_0^1 F'(t)\, \mathrm{d}t$, and it remains to compute F'. Since $\int \partial_t p_t = 0$,

$$F'(t) = \int \partial_t p_t\, \log\frac{p_t}{q_t} - \int \frac{p_t}{q_t}\, \partial_t q_t.$$

Substituting Lemma 2 for p_t and q_t and integrating by parts,

$$\int \nabla\cdot(p_t v_t^p) \log\frac{p_t}{q_t} = -\int p_t\, v_t^p \cdot \nabla \log\frac{p_t}{q_t},$$
$$\int \frac{p_t}{q_t}\, \nabla\cdot(q_t v_t^q) = -\int q_t\, v_t^q \cdot \nabla\frac{p_t}{q_t} = -\int p_t\, v_t^q \cdot \nabla \log\frac{p_t}{q_t},$$

so that

$$F'(t) = -\mathbb{E}_{z\sim p_t}\big[\big(v_t^p(z) - v_t^q(z)\big)\cdot\big(\nabla \log p_t(z) - \nabla \log q_t(z)\big)\big]\,.$$

By Lemma 1, $\nabla \log p_t - \nabla \log q_t = \frac{1-t}{t}(v_t^p - v_t^q)$, since the term $-z/t$ is common to both. Therefore $F'(t) = -\frac{1-t}{t}\, \mathbb{E}_{z\sim p_t}\, \|v_t^p(z) - v_t^q(z)\|^2$, and integrating over $[0,1]$ gives Equation (7). □

The weight $(1-t)/t$ diverges as $t \to 0$, and the integral is finite because both fields converge to the identity at the boundary. Substituting Equation (6) instead expresses the same integral as a score-difference integral with weight $t/(1-t)$; the velocity form is the one whose integrand is bounded at every t, which is why our model predicts velocities.

B.3 MUTUAL INFORMATION

We use the notation of Section 2.1: $z_0 = (x_0, y_0) \sim p_{XY}$, $\epsilon = (\epsilon_X, \epsilon_Y)$, $x_t = (1-t)x_0 + t\epsilon_X$, $y_t = (1-t)y_0 + t\epsilon_Y$, $z_t = (x_t, y_t)$, the joint field $v_t(z)$ with blocks $v_t(z)|_X$ and $v_t(z)|_Y$, and the conditional fields $v_t(x \mid y_0)$ and $v_t(y \mid x_0)$ of $p_{X|Y=y_0}$ and $p_{Y|X=x_0}$. In addition, $v_t^X(x) = \mathbb{E}[X_0 - \epsilon_X \mid X_t = x]$ and $v_t^Y(y)$ denote the velocity fields of the marginals p_X and p_Y. All expectations below are over x_0, y_0, ϵ.

Lemma 3 (Field of the product of marginals)**.** *The velocity field of $p_X \otimes p_Y$ is $(x, y) \mapsto \big(v_t^X(x), v_t^Y(y)\big)$.*

Proof. Under $p_X \otimes p_Y$ the pairs (X_0, ϵ_X) and (Y_0, ϵ_Y) are independent, so conditioning $X_0 - \epsilon_X$ on (X_t, Y_t) is the same as conditioning it on X_t alone, and symmetrically for Y. □

Proposition 1 (Product and conditional forms)**.**

$$\mathrm{I}(X;Y) = \int_0^1 \frac{1-t}{t}\, \mathbb{E}\Big[\big\|v_t(z_t)|_X - v_t^X(x_t)\big\|^2 + \big\|v_t(z_t)|_Y - v_t^Y(y_t)\big\|^2\Big]\, \mathrm{d}t, \tag{8}$$

$$\mathrm{I}(X;Y) = \int_0^1 \frac{1-t}{t}\, \mathbb{E}\Big[\big\|v_t(x_t \mid y_0) - v_t^X(x_t)\big\|^2\Big]\, \mathrm{d}t = \int_0^1 \frac{1-t}{t}\, \mathbb{E}\Big[\big\|v_t(y_t \mid x_0) - v_t^Y(y_t)\big\|^2\Big]\, \mathrm{d}t. \tag{9}$$

Proof. Equation (8) is Theorem 1 with $p = p_{XY}$ and $q = p_X \otimes p_Y$, using Lemma 3 and splitting the squared norm into its two blocks. For Equation (9), $\log \frac{p_{XY}(x,y)}{p_X(x)p_Y(y)} = \log \frac{p_{X|Y}(x|y)}{p_X(x)}$ gives $\mathrm{I}(X;Y) = \mathbb{E}_{y_0} \mathrm{KL}\left[p_{X|Y=y_0} \,\|\, p_X\right]$; applying Theorem 1 to each pair $(p_{X|Y=y_0}, p_X)$ and averaging over y_0 gives the first expression, and the second follows by symmetry. □

Theorem 2 (Joint-versus-conditional form)**.** *With one perturbation ϵ shared by the three fields, Equation (3) holds:*

$$\mathrm{I}(X;Y) = \int_0^1 \frac{1-t}{t}\, \mathbb{E}\Big[\|v_t(z_t)|_X - v_t(x_t \mid y_0)\|^2 + \|v_t(z_t)|_Y - v_t(y_t \mid x_0)\|^2\Big]\, \mathrm{d}t.$$

Proof. Fix t and consider the X block. The three fields are conditional expectations of the same variable $X_0 - \epsilon_X$ under three conditionings:

$$v_t^X(x_t) = \mathbb{E}[X_0 - \epsilon_X \mid x_t], \quad v_t(z_t)|_X = \mathbb{E}[X_0 - \epsilon_X \mid x_t, y_t], \quad v_t(x_t \mid y_0) = \mathbb{E}[X_0 - \epsilon_X \mid x_t, y_0].$$

Since ϵ_Y is independent of (x_0, ϵ_X, y_0), the point $y_t = (1-t)y_0 + t\epsilon_Y$ carries no information about $X_0 - \epsilon_X$ beyond (x_t, y_0), so by the tower property

$$v_t(z_t)|_X = \mathbb{E}[v_t(x_t \mid y_0) \mid x_t, y_t] \qquad \text{and} \qquad v_t^X(x_t) = \mathbb{E}[v_t(z_t)|_X \mid x_t]\,.$$

That is, $v_t(z_t)|_X$ is the orthogonal projection of $v_t(x_t \mid y_0)$ onto the functions of (x_t, y_t), and $v_t^X(x_t)$ is the projection of both onto the functions of x_t, so the two increments are orthogonal and

$$\mathbb{E}\,\big\|v_t(x_t \mid y_0) - v_t^X(x_t)\big\|^2 = \mathbb{E}\,\|v_t(x_t \mid y_0) - v_t(z_t)|_X\|^2 + \mathbb{E}\,\big\|v_t(z_t)|_X - v_t^X(x_t)\big\|^2\,.$$

The same identity holds for the Y block. Adding the two blocks, multiplying by $(1-t)/t$, and integrating, the left sides are the two expressions of Equation (9), each equal to $\mathrm{I}(X;Y)$, and the last terms sum to the integrand of Equation (8), also equal to $\mathrm{I}(X;Y)$. The integral of the middle terms is therefore $\mathrm{I}(X;Y) + \mathrm{I}(X;Y) - \mathrm{I}(X;Y) = \mathrm{I}(X;Y)$. □

Remark 1 (Shared noise). The projection argument requires the joint and the conditional field to be evaluated at the same noised block, with the same ϵ_X in $v_t(z_t)|_X$ and $v_t(x_t \mid y_0)$ and the same ϵ_Y in the Y term. With independent perturbations the increments are no longer orthogonal and the integrand no longer averages to the mutual information.

B.4 Conditional variant: a discrete input as clean evidence

An equivalent form for estimating MI writes $\mathrm{I}(X;Y) = \mathbb{E}_y \mathrm{KL}\left[p_{X|Y=y} \parallel p_X\right]$ and compares the conditional velocity of X given Y to the marginal velocity of X. When both distributions are continuous we use the Equation (3) form, because a single indicator-conditioned field yields all required partial velocities without a separate marginal model. When the input is discrete, the conditional form becomes a finite sum and we can use the estimator in Appendix G. Let X take one of m values $x_1, \ldots, x_m$ with weights $p = (p_1, \ldots, p_m)$ and let Y be a continuous response. Then

$$\mathrm{I}(X;Y) = \sum_{i=1}^{m} p_i \, D_i, \qquad D_i = \mathrm{KL}\left[P_{Y|x_i} \parallel \bar{P}\right], \qquad \bar{P} = \sum_{j=1}^{m} p_j \, P_{Y|x_j}, \tag{10}$$

and each divergence is the velocity-form KL of Equation (7) applied to the response alone,

$$D_i = \int_0^1 \frac{1-t}{t} \, \mathbb{E}_{y_0 \sim P_{Y|x_i}, \epsilon} \left\| v_t(y_t|x_i) - \bar{v}_t(y_t) \right\|^2 \mathrm{d}t, \qquad y_t = (1-t)y_0 + t\epsilon. \tag{11}$$

Only Y is noised: every query holds the input block at the atom x_i as clean evidence through the noising indicator (input coordinates clean, response coordinates noised), only the response block of the output is read, and no velocity field is needed for the discrete coordinate.

Two contexts from one field. The two fields in Equation (11) are the same model call, with the same indicator, at the same evaluation points, bound to two different contexts. Bound to a *joint* context of clean (x, y) rows, the evidence x_i selects the conditional $P_{Y|x_i}$, and the call returns its velocity $v_t^{x_i}$. Bound to a *shuffled* context, the same call returns $\bar{v}_t$. The shuffled context is built row by row from two independent draws from p: the first draw selects an input value and the row takes a response from the pool of that value, the second draw overwrites the input column. Input and response are therefore independent in the context, so the evidence carries no information about the response, and the response marginal of the context is $\bar{P}$ by construction, whatever p is. Drawing the shuffled context from the same pooled rows as the joint context keeps part of the finite-context sampling noise common to the two fields. The joint context is stratified uniformly over the atoms, since the conditionals do not depend on p; the shuffled context follows p which determines $\bar{P}$ when p changes.

C Estimation algorithm

Algorithm 1 lists the mutual information estimation procedure of Section 2.1: a disjoint context/evaluation split, one cached context encoding, three velocity queries per (point, time) pair with shared noise, and the weighted average of the block-wise velocity differences of Equation (4).

Algorithm 1 ALICE mutual information estimation

Require: N joint samples $z = (x, y)$; frozen ALICE v_θ; time draws per point n_t; indicators $m_{XY} = (\mathbb{1}_X, \mathbb{1}_Y)$, $m_X = (\mathbb{1}_X, \mathbb{0}_Y)$, $m_Y = (\mathbb{0}_X, \mathbb{1}_Y)$

split the samples into a disjoint *context* set C of size n and *evaluation* set $\mathcal{D}_{\text{eval}}$; fit a coordinate-wise copula map on C and apply it to both sets
encode C once and cache its keys and values ▷ reused by every query below
for each evaluation point $z_0 = (x_0, y_0) \in \mathcal{D}_{\text{eval}}$ and each of n_t draws $t \sim \mathcal{U}[0,1]$ **do**
  draw one $\epsilon \sim \mathcal{N}(0, I)$; set $z_{\text{full}} = (1-t)z_0 + t\epsilon$ ▷ shared by the three queries
  $z_X \leftarrow m_X \odot z_{\text{full}} + (1 - m_X) \odot z_0$; $z_Y \leftarrow m_Y \odot z_{\text{full}} + (1 - m_Y) \odot z_0$
  $v_{\text{full}} \leftarrow v_\theta(z_{\text{full}}, t, m_{XY}; C)$ ▷ three queries to the cached context
  $v_X \leftarrow v_\theta(z_X, t, m_X; C)$; $v_Y \leftarrow v_\theta(z_Y, t, m_Y; C)$
  $g \leftarrow \frac{1-t}{t}\left[\left\|(v_{\text{full}} - v_X)|_X\right\|^2 + \left\|(v_{\text{full}} - v_Y)|_Y\right\|^2\right]$
end for
return $\widehat{\mathrm{I}}(X;Y) = \frac{1}{N_{\text{MC}}} \sum g$ over the $N_{\text{MC}} = |\mathcal{D}_{\text{eval}}| \, n_t$ pairs ▷ Equation (4)

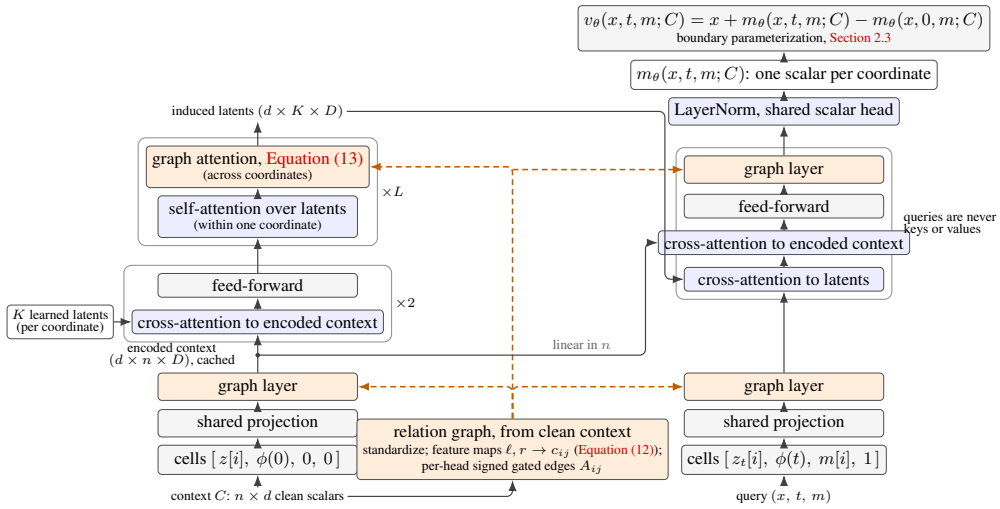


Figure 7: One forward pass of ALICE. Clean context values determine the relation graph, whose gated edges A_{ij} condition every graph layer (dashed). Context and query cells pass through the same shared projection and input graph layer; K learned latents per coordinate read the encoded context twice through cross-attention, and L blocks alternate self-attention among one coordinate's latents with graph attention across coordinates. A query decodes in one pass, cross-attending to the latents for the global summary and to the encoded context for local detail, and a shared scalar head produces one output per coordinate; two head evaluations, at times t and 0, form the velocity through the boundary parameterization. The context side, left on the figure, is encoded once per context and cached; every interaction with the n context samples is linear in n.

D ALICE DETAILS

This section specifies the ALICE architecture summarized in Section 2.3: the token layout and time conditioning, the relation graph and its attention rule, the induced context bottleneck, the boundary parameterization, and the model family. Figure 7 shows one forward pass through these components.

D.1 ARCHITECTURE

Conditioning and caching. Queries interact with the network only through cross-attention: they are not used as keys or values. Three properties follow. 1) Context representations never depend on queries, so a context is encoded once per distribution, cached, and reused by every velocity evaluation. 2) Each query's output is a function of $(z_t, t, m; C)$ alone, and does not depend on other queries that might be added or permuted: hence, many queries are scored in one pass and the boundary parameterization of Section 2.3 is exact. 3) The context-validity mask excludes padded context samples from every attention over the context, so a padded context is equivalent to a physically truncated one and the same weights serve any context length. Every token is embedded by a shared projection, but no channel of the embedding encodes the index of the sample or of the coordinate a token comes from: a context is an exchangeable set of samples and a sample is an unordered set of coordinates, so the network carries no positional information along either axis. The noising indicator m plays no role in attention; it is used only as an input channel of the query tokens.

Per-coordinate tokens and time conditioning. The joint width d is not fixed a priori. Every scalar coordinate of every sample becomes one token [value, $\phi(t)$, $m[i]$, type], lifted to width D by one shared projection; context and query tokens then pass through one shared input graph layer before any processing (Figure 7, bottom). All parameters live in coordinate-shared maps: the token projection, the attention and feed-forward weights, the latent bank, and a scalar output head. Changing d therefore changes only the number of tokens per sample, and the same model weights can be used at any joint width. We use sinusoidal time features $\phi(t) = [t, \sin(2^k t), \cos(2^k t)]_{k<F}$, and the noising-indicator entry $m[i] \in \{0, 1\}$ encodes partial observation: a 1 marks a coordinate that follows the interpolant at time t and is to be predicted, a 0 indicates a coordinate held clean at its observed value as evidence.

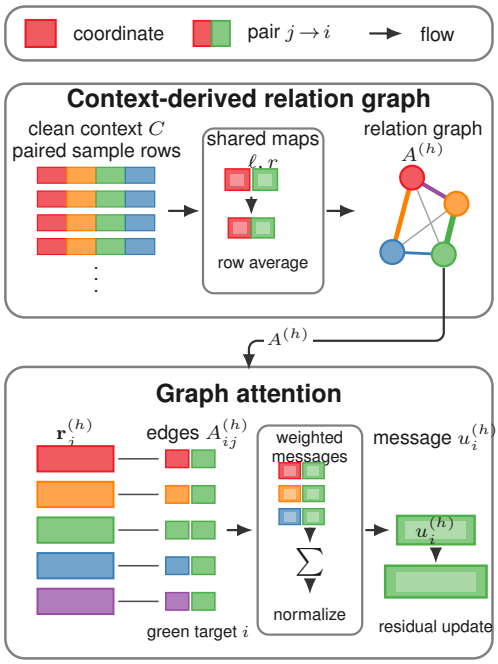


Figure 8: ALICE relation-graph construction and coordinate mixing. Node and block colors identify coordinates, and each two-color block denotes a source-target pair $j \to i$. Top: shared feature maps ℓ, r aggregate paired clean-context rows into signed, gated relation weights $A_{ij}^{(h)}$. Bottom: for the green target i, each source representation $\mathbf{r}_j^{(h)}$ is weighted by its incoming edge, and the center module sums and normalizes these contributions to produce $u_i^{(h)}$ for the residual update.

Context tokens carry zero time features and an all-clean indicator. The indicator channel lets a single model produce the three partially noised velocities of Equation (3); the boundary parameterization evaluates the network at times t and 0 with the same indicator m.

The relation graph. The cross-coordinate mechanism must represent which coordinates depend on which, with what sign, possibly through non-monotone relations, all varying from distribution to distribution. A learned $d \times d$ interaction parameter would be tied to one dimension and one dependence pattern, and softmax attention across coordinates produces weights that are dense, nonnegative, and sum to one, so independent coordinates would still exchange information. ALICE instead measures the dependence structure from the clean context and uses the result as a weighted graph over coordinates, recomputed once per forward pass whenever the context changes; the resulting edges condition every graph layer in Figure 7 (dashed). Figure 8 summarizes the construction.

To describe pairwise dependence, we measure covariance between learned nonlinear features, a principle also used in kernel dependence measures (Gretton et al., 2005). Let $\hat{z}^{(k)}[i]$ be coordinate i of clean context sample k, standardized over the context. Two learned maps $\ell, r : \mathbb{R} \to \mathbb{R}^R$, each shared across coordinates and context samples, take this single scalar as input and output R nonlinear features. Subtracting each feature's context mean gives $\tilde{\ell}_i^{(k)} = \ell(\hat{z}^{(k)}[i]) - \frac{1}{n}\sum_{k'=1}^{n} \ell(\hat{z}^{(k')}[i])$, and likewise $\tilde{r}_i^{(k)}$. The descriptor of coordinates i and j is

$$c_{ij} = \frac{1}{2n} \sum_{k=1}^{n} \left(\tilde{\ell}_i^{(k)} \odot \tilde{r}_j^{(k)} + \tilde{r}_i^{(k)} \odot \tilde{\ell}_j^{(k)} \right) \in \mathbb{R}^R. \tag{12}$$

Each component of c_{ij} is an average of two empirical feature covariances, with the common index k preserving the joint observations and symmetrization giving $c_{ij} = c_{ji}$. With identity feature maps, Equation (12) reduces to the empirical correlation of the standardized coordinates; learned maps expose dependence, such as $z[j] \approx z[i]^2$, that correlation misses. Centering makes the population descriptor vanish under independence, since the expected product of centered features then factorizes,

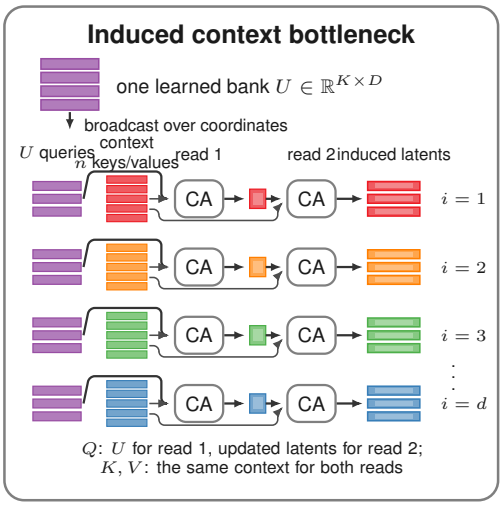


Figure 9: Induced context bottleneck. The learned bank U is broadcast across coordinates. For each coordinate, the first cross-attention layer uses U as queries and that coordinate's context tokens as keys and values; the second uses the updated latents as queries and the same context tokens as keys and values, producing K context-specific latent vectors.

although finite contexts introduce sampling fluctuations. Permuting all context samples together leaves the descriptor invariant, while permuting one coordinate's values independently changes the empirical joint and therefore the graph.

The descriptor is shared by all attention heads. For graph head h, learned projection vectors $w_s^{(h)}, w_g^{(h)} \in \mathbb{R}^R$ and scalar biases $b_s^{(h)}, b_g^{(h)}$ convert it into a signed, gated edge. With σ the sigmoid and $\mathbf{r}_j^{(h)}$ the projected representation of coordinate j at the same context, latent, or query position, the edge and the aggregated message are

$$A_{ij}^{(h)} = \tanh\Big((w_s^{(h)})^\top c_{ij} + b_s^{(h)}\Big)\,\sigma\Big((w_g^{(h)})^\top c_{ij} + b_g^{(h)}\Big), \qquad u_i^{(h)} = \frac{\sum_{j\neq i} A_{ij}^{(h)} \mathbf{r}_j^{(h)}}{\max\Big(1, \sum_{j\neq i} |A_{ij}^{(h)}|\Big)}, \tag{13}$$

followed by an output projection and the usual residual and feed-forward updates. The signed factor allows additive or subtractive contributions, while the gate controls their magnitude. Normalizing by absolute edge mass bounds the aggregate contribution, and the lower bound of one preserves small updates when all edges are weak. There are no self-edges. Gates are initialized nearly closed, so training starts from an independence prior and opens edges only where the context provides evidence of dependence; exact disconnection under independence is not enforced. The feature maps and edge projections are learned through the velocity objective, so the edges represent pairwise associations useful for prediction without imposing a conditional-independence interpretation. Sharing these maps across coordinates keeps the parameter count independent of d and makes the graph equivariant to coordinate permutations.

Induced context bottleneck. The model compresses each coordinate's n context tokens into K induced latents and runs its depth on the latents at a cost that is independent of n (Figure 7, left tower). In other words, a shared bank of K learned vectors reads the encoded context through two cross-attentions, each linear in n, and the deep blocks then alternate self-attention among one coordinate's latents, refining that coordinate's summary of the context, with graph attention from Equation (13), sharing the summaries across coordinates. A query decodes through two complementary mechanisms (Figure 7, right tower): cross-attention to the latents provides the global summary, and a final cross-attention to the encoded context, linear in n, retrieves the local detail near the query that a K-vector summary cannot retain; a last graph layer and a shared scalar head produce one output per coordinate. The remaining cost is the $d \times d$ relation graph, which is favorable in the long-context, moderate-dimension regime of MI estimation.

Boundary parameterization. The estimator multiplies squared velocity differences by $(1-t)/t$, which diverges as $t \to 0$. Since squared differences cannot be negative, any violation of the boundary condition $v_0(z) = z$ stated in Section 1 becomes systematic positive bias where the weight is largest. ALICE satisfies this condition by construction by adopting the parametrization described in Wang et al. (2026); Hu et al. (2025).

Model family. We instantiate ALICE at a range of sizes that share the number of induced latents K, the relation-feature width R, and the time-feature resolution, so that model size affects only the backbone capacity, without changing the context bottleneck or the graph statistic. Parameter counts are independent of the joint width and the context length, and a trained checkpoint is exported as a self-contained model (weights, configuration, and source), usable at any joint width without modification.

The Small and Base configurations are listed in Table 1. ALICE Base is the checkpoint reported in Section 3, and Appendix F compares the Small and Base checkpoints on the benchmark.

Table 1: The two ALICE variants, with model configuration values and parameter counts for each preset.

Config name	ALICE Small	ALICE Base
Model hidden dimension	384	768
Layers	5	6
Attention heads	8	12
Feed-forward dimension	1536	3072
Time frequencies	16	16
Dropout	0.1	0.1
Induced latents	128	128
Relation features	16	16
Parameters	19,296,113	85,200,633

E TRAINING AND IMPLEMENTATION DETAILS

This section presents the reference implementation of ALICE.

E.1 THE TRAINING CORPUS

A corpus episode is one synthetic joint distribution over $z = (x, y) \in \mathbb{R}^d$, generated from a seed, with the fixed split $s = \lfloor d/2 \rfloor$: coordinates $[0, s)$ are X and $[s, d)$ are Y. Each episode is stored as a clean point pool of 2176 samples in single precision, together with a few scalar metadata fields; normalization, noising, indicator sampling, and targets are computed at train time.

Composition. The corpus covers the joint widths $\{2, 3, 4, 5, 6, 8, 10, 12, 16, 20, 25, 32, 50, 100\}$ with 70,000 episodes per width, drawn from four families: copula mixtures, latent warps, manifolds, and nonparametric regressions, with probabilities 0.30, 0.25, 0.25, and 0.20. A further copula-only share brings the copula fraction of the whole corpus to about 0.40, and part of the corpus enables the two geometric modifications described below, same-sign factor covariances and the plane-rotation warp.

Copula mixtures. Between 1 and 60 Gaussian or Student-t components with random weights and means. Each component draws a low-rank covariance $\Sigma = WW^\top + D$ of random rank, converted to a correlation and rescaled per coordinate; with probability 0.3 it instead draws a sparse correlation with a few disjoint $X_i \leftrightarrow Y_i$ pairs whose strength is coherent within an episode; and with probability $\frac{1}{2}$ the $X \leftrightarrow Y$ cross-block of every component is scaled down, to zero half the time, which produces weakly dependent and independent joints. Most sampled pools are then passed through an additive-coupling bijection (Dinh et al., 2017), either within each block, which preserves $I(X;Y)$, or across a random coordinate partition, which leaves it unknown.

Latent warps. A mixture of anisotropic Gaussians whose means lie along a random curve is standardized and pushed through a few random layers, each an additive coupling shift, an elementwise sinusoidal fold, or a rotation. The fold is non-injective, so $I(X;Y)$ is unknown by construction.

Nonparametric regressions. The input is Gaussian, or a two-component mixture, and the response is a random Fourier-feature function of the input plus Gaussian noise of random scale, which provides a controlled noise floor and a smooth nonlinear conditional mean.

Manifolds. The pool lies on a low-dimensional curved support, a curve or a surface winding around the origin, thickened by transverse Gaussian noise of random scale and rotated at random. At small noise the support is near-singular, which is the regime where a velocity field must resolve a thin set.

Plane-rotation warp. An MI-preserving diffeomorphism rotates randomly chosen coordinate planes of a block by an angle that grows with the block norm, occasionally followed by a monotone radial stretch. Each rotation preserves the block norm, so $I(X;Y)$ is unchanged, and the warp acts on a point cloud, so it applies to every family.

Same-sign factor covariances. The low-rank draw above has sign-symmetric loadings, so joints in which every coordinate pair is positively correlated, a common structure in measured data with a shared latent factor, have vanishing probability under it. Part of the corpus therefore draws equicorrelated or positive low-rank covariances instead.

E.2 Batch construction

For each batch, the procedure (i) samples one context length shared by all its distributions (for variable-context training), (ii) samples disjoint context/query samples from each pool, and (iii) applies Gaussian-copula softrank normalization: each marginal is mapped to $\mathcal{N}(0,\,1)$ through the empirical CDF *fit on the context* and applied out-of-sample to the query. Batches are dimension-homogeneous: each batch is drawn from a single joint width. Variable context length is realized either by truncating to the sampled length or by hiding context samples behind the context-validity mask; the two are equivalent (see also Section 2.3).

E.3 Objective, noising indicators, optimizer

The loss is the masked velocity MSE defined in Equation (5), supervised only on noised coordinates. Each query draws its own time $t \sim \mathcal{U}[0,1]$ and noise $\epsilon \sim \mathcal{N}(0,\,I)$, and the target $z_0 - \epsilon$ is available exactly because the training loop draws ϵ, t, and m itself; no ground-truth density or MI values are required for training at any point.

The per-query noising-indicator mixture is: all-noised with prob. 0.35; an $X \mid Y$ or $Y \mid X$ block pattern with prob. 0.30 (split evenly); otherwise a per-coordinate $\mathrm{Bernoulli}(0.5)$ indicator (all-zero draws fall back to all-noised). The mixture covers the three indicator patterns the estimator queries at inference (Equation (3)) and, through the random subsets, general partial observation. The block patterns use the fixed split $s = \lfloor d/2 \rfloor$. Training otherwise operates on the whole vector z: the $X{:}Y$ partition is used in training only through those block masks and through the corpus's block-structured couplings (decoupling and per-block flows, also at s), and the specific partition otherwise appears only at output time.

Each training step draws 128 distributions from the corpus and one clean context from each. The context length is sampled uniformly per batch between 128 and the training window (by truncation, or equivalently by masking, Appendix E.2), so one set of weights is trained for every context length up to that window; longer contexts are extrapolation (Appendix F). The optimizer is AdamW with $\beta_1{=}0.9$, $\beta_2{=}0.95$, and weight decay 0.01, with a linear warmup over 100 steps and gradient norms clipped at 1.0. Training is executed in `bf16` mixed precision with compiled kernels on two data-parallel replicas (DDP), with gradient accumulation setting the effective batch size. Training proceeds in two phases. The first runs $500{,}000$ steps with a context window of 1024 samples and a cosine decay of the learning rate to zero after the warmup. The second starts from the first-phase weights with a fresh optimizer state and runs $60{,}000$ steps with a context window of 2048 samples, holding the learning rate constant after the warmup. The peak learning rate is $3 \cdot 10^{-4}$ for ALICE Base and 10^{-3}

for ALICE Small in both phases; the per-device batch is 16 distributions with 4 accumulation steps, except for ALICE Base in the second phase, which uses 8 with 8.

E.4 HARDWARE FOR TRAINING AND INFERENCE

We train our ALICE variants using 2x H200 GPUs: ALICE-Small requires 2 days and 22 hours (about 8,674 optimizer steps per hour) whereas ALICE-Base requires 5 days and 10 hours (about 4,300 optimizer steps hour) of training. As a comparison, our understanding is that InfoAtlas (Hu et al., 2026) requires 2 weeks of training on 16x H800 GPUs.

For inference, we use a single H200 GPU in all our experiments.

F GROUND-TRUTH BENCHMARK: DETAILS AND PER-TASK RESULTS

This Section completes Section 3: it specifies the inference settings, reports the aggregate (Table 2), joint-width (Table 3), and per-task (Table 4) results at the three matched budgets.

Inference settings. Every task provides precomputed samples and a closed-form ground-truth MI. We use three sample budgets $N \in \{1000, 5000, 10000\}$ for all methods. ALICE splits each budget into 64 query samples and a context of the remaining $N - 64$ clean samples; Algorithm 1 averages the velocity-difference integrand over the 64 query samples and n_t=64 time draws per query sample. Every reported number is a mean over eight independent context draws; where a spread is given, it is the sample standard deviation over the draws. Training samples the context length uniformly between 128 and the training window, 2048 samples in the final phase (Appendix E.3), so the contexts at the 5000 and 10000 budgets are extrapolation beyond the training window; the context attention is permutation invariant and uses no positional encoding, so the model accepts these longer contexts, and Table 3 reports how each model size behaves there. Per-coordinate monotone transforms (`normal_cdf`, `half_cube`, `asinh`) are absorbed by the rank-based copula normalization applied at inference; Figure 2 groups them with their base tasks and Table 4 lists them separately. InfoAtlas conditions on the same contexts. Competitor numbers are five-seed means: the neural estimators are trained on the N samples of each individual task (including per-task hyperparameter tuning), and the classic estimators are fit on them.

Aggregate accuracy. Table 2 reports the MAE of every estimator over the suite at the three budgets. ALICE Base has the lowest error at each budget among the estimators of Figure 2, and its lead is largest at 1k samples, where every neural estimator is above 0.24 nats and the best classic estimator, CCA, is at 0.195. Table 4 reports every estimate at 10k samples.

Joint width. Table 3 splits the error of both ALICE sizes and InfoAtlas between the 33 tasks of joint width at most 10 and the 7 tasks of width 50 and 100. ALICE Base improves with the context in both groups, and most in the wide one, from 0.153 to 0.086 nats. ALICE Small improves on the narrow tasks, from 0.081 to 0.074 nats, and degrades on the wide ones, from 0.197 to 0.241, so its aggregate error is flat in N. InfoAtlas is at 0.10 to 0.12 nats on the narrow tasks, where its per-width networks apply, and at 1.0 nats on the wide tasks, where its sliced fallback outputs 0.02 nats on the five sparse tasks and 0.46 on the two dense ones against ground truths of 1.02 to 1.62.

Error analysis. At N=10000, 35 of the 40 tasks fall within 0.1 nats of ground truth for ALICE Base and the mean signed error is -0.005 nats, so the aggregate measure is not influenced by a global bias. Two groups impact the results. First, the spiral embeddings are under-estimated by 0.43 and 0.44 nats at joint widths 6 and 10 and by 0.18 nats at width 50, the largest errors ALICE experiences in the suite. Second, the dense multinormal tasks are over-estimated, by 0.23 nats at joint width 100 and 0.15 at width 50, growing with width. ALICE Small shares both failure modes with larger magnitudes: it under-estimates the width-10 spiral by 0.47 nats and over-estimates the width-100 dense multinormal by 0.44, and it also over-estimates three of the five sparse width-50 tasks by 0.24 nats each.

Table 2: Mean absolute error in nats over the 40 tasks of the Czyż benchmark at matched budgets of 1k, 5k, and 10k samples per task. ALICE and InfoAtlas condition on a context of $N - 64$ samples from that budget, and their cells give the mean and sample standard deviation over eight independent context draws; neural estimators are trained per distribution on that number of samples and classic estimators are fit on it, both as five-seed means. "> 10" marks a diverged estimator.

Estimator	1k	5k	10k
Foundation models			
ALICE Base	0.092 ± 0.004	0.063 ± 0.001	0.060 ± 0.001
ALICE Small	0.101 ± 0.005	0.103 ± 0.002	0.104 ± 0.001
InfoAtlas	0.253 ± 0.004	0.271 ± 0.005	0.276 ± 0.001
Neural estimators			
MINDE–C	0.353	0.070	0.065
MINE	0.248	0.117	0.108
InfoNCE	0.887	0.123	0.083
D-V	1.229	0.279	0.144
NWJ	> 10	1.261	0.130
Classic estimators			
KSG	0.288	0.240	0.219
LNN	4.864	4.966	4.970
CCA	0.195	0.192	0.181

Table 3: Czyż benchmark accuracy by joint width against the sample budget N for both ALICE sizes and InfoAtlas. "dims ≤ 10" aggregates the 33 tasks of joint width at most 10 and "dims $50/100$" the 7 wider ones. Cells give the mean and sample standard deviation of the per-draw MAE over eight independent context draws. The ALICE training window is 2048 samples, so the rows at 5000 and 10000 are context extrapolation.

Model	N	All (40)	dims ≤ 10 (33)	dims $50/100$ (7)
ALICE Base	1000	0.092 ± 0.004	0.079 ± 0.004	0.153 ± 0.008
	5000	0.063 ± 0.001	0.057 ± 0.002	0.090 ± 0.004
	10000	0.060 ± 0.001	0.054 ± 0.001	0.086 ± 0.002
ALICE Small	1000	0.101 ± 0.005	0.081 ± 0.005	0.197 ± 0.017
	5000	0.103 ± 0.002	0.077 ± 0.001	0.228 ± 0.009
	10000	0.104 ± 0.001	0.074 ± 0.002	0.241 ± 0.006
InfoAtlas	1000	0.253 ± 0.004	0.100 ± 0.004	0.978 ± 0.008
	5000	0.271 ± 0.005	0.117 ± 0.006	0.998 ± 0.005
	10000	0.276 ± 0.001	0.123 ± 0.003	1.000 ± 0.006

Table 4: Per-task MI estimates on the Czyż et al. (2023) suite against ground truth (GT), in nats, with every estimator at a budget of 10k samples per task. Cell shading encodes the signed bias of the estimate: blue for over-estimation, red for under-estimation, with saturation growing linearly up to a bias of 0.6 nats. Competitor rows report five-seed means. The ALICE rows use zero-shot in-context estimation with a frozen model, and the ALICE and InfoAtlas rows are means over eight independent context draws. Abbreviations: *Mn* multinormal, *St* Student-t, *Nm* normal, *Hc* half-cube, *Sp* spiral.

dist	Asinh @ *St* 1 × 1 (dof=1)	Asinh @ *St* 2 × 2 (dof=1)	Asinh @ *St* 3 × 3 (dof=2)	Asinh @ *St* 5 × 5 (dof=2)	Bimodal 1 × 1	Bivariate *Nm* 1 × 1	*Hc* @ Bivariate *Nm* 1 × 1	*Hc* @ *Mn* 25 × 25 (2-pair)	*Hc* @ *Mn* 3 × 3 (2-pair)	*Hc* @ *Mn* 5 × 5 (2-pair)	*Mn* 2 × 2 (2-pair)	*Mn* 2 × 2 (dense)	*Mn* 25 × 25 (2-pair)	*Mn* 25 × 25 (dense)	*Mn* 3 × 3 (2-pair)	*Mn* 3 × 3 (dense)	*Mn* 5 × 5 (2-pair)	*Mn* 5 × 5 (dense)	*Mn* 50 × 50 (dense)	*Nm* CDF @ Bivariate *Nm* 1 × 1	*Nm* CDF @ *Mn* 25 × 25 (2-pair)	*Nm* CDF @ *Mn* 3 × 3 (2-pair)	*Nm* CDF @ *Mn* 5 × 5 (2-pair)	*Sp* @ *Mn* 25 × 25 (2-pair)	*Sp* @ *Mn* 3 × 3 (2-pair)	*Sp* @ *Mn* 5 × 5 (2-pair)	*Sp* @ *Nm* CDF @ *Mn* 25 × 25 (2-pair)	*Sp* @ *Nm* CDF @ *Mn* 3 × 3 (2-pair)	*Sp* @ *Nm* CDF @ *Mn* 5 × 5 (2-pair)	*St* 1 × 1 (dof=1)	*St* 2 × 2 (dof=1)	*St* 2 × 2 (dof=2)	*St* 3 × 3 (dof=2)	*St* 3 × 3 (dof=3)	*St* 5 × 5 (dof=2)	*St* 5 × 5 (dof=3)	Swiss roll 2 × 1	Uniform 1 × 1 (additive noise=.1)	Uniform 1 × 1 (additive noise=.75)	Wiggly @ Bivariate *Nm* 1 × 1
GT	0.22	0.43	0.29	0.45	0.41	0.41	0.41	1.02	1.02	1.02	1.02	0.29	1.02	1.29	1.02	0.41	1.02	0.59	1.62	0.41	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.02	0.22	0.43	0.19	0.29	0.18	0.45	0.30	0.41	1.71	0.33	0.41
Foundation models																																										
ALICE Base	0.26	0.42	0.30	0.44	0.42	0.42	0.42	1.03	1.07	1.09	1.06	0.29	1.03	1.44	1.07	0.43	1.10	0.64	1.85	0.42	1.03	1.08	1.09	0.84	0.59	0.58	1.00	0.96	1.01	0.26	0.42	0.21	0.30	0.19	0.45	0.29	0.48	1.63	0.31	0.42
ALICE Small	0.23	0.46	0.29	0.37	0.49	0.49	0.48	1.27	1.10	1.06	1.15	0.34	1.26	1.62	1.10	0.48	1.05	0.71	2.06	0.48	1.26	1.11	1.06	0.89	0.69	0.55	1.09	1.03	1.01	0.23	0.45	0.20	0.30	0.17	0.38	0.23	0.55	1.85	0.35	0.48
InfoAtlas	0.18	0.33	0.22	0.34	0.40	0.39	0.40	0.02	0.98	0.97	0.96	0.28	0.02	0.46	0.98	0.39	0.96	0.56	0.46	0.39	0.02	0.98	0.96	0.02	0.49	0.61	0.02	0.87	0.88	0.04	0.07	0.15	0.22	0.14	0.34	0.23	0.10	1.10	0.17	0.40
Neural estimators																																									
MINDE–C	0.19	0.39	0.26	0.41	0.40	0.40	0.40	0.97	0.98	0.96	1.01	0.29	1.04	1.20	0.99	0.43	0.99	0.61	1.51	0.39	1.01	0.93	0.92	0.89	0.81	0.79	1.01	0.91	0.91	0.10	0.17	0.17	0.23	0.17	0.39	0.28	0.25	1.62	0.29	0.40
MINE	0.22	0.38	0.20	0.30	0.42	0.42	0.42	0.87	0.99	0.96	1.01	0.30	0.91	1.25	1.02	0.41	0.98	0.58	1.60	0.41	0.82	0.91	0.88	0.74	0.66	0.62	0.81	0.88	0.86	0.02	0.02	0.10	0.11	0.12	0.15	0.17	0.40	1.66	0.31	0.42
InfoNCE	0.22	0.40	0.22	0.35	0.41	0.42	0.42	0.82	1.00	0.97	1.02	0.29	0.88	1.25	1.01	0.41	0.99	0.58	1.56	0.41	0.77	0.96	0.91	0.71	0.75	0.69	0.76	0.93	0.88	0.09	0.24	0.16	0.23	0.14	0.37	0.21	0.40	1.69	0.32	0.42
D-V	0.22	0.39	0.23	0.35	0.41	0.42	0.41	0.83	1.00	0.97	1.02	0.29	0.89	1.26	1.01	0.41	0.98	0.58	1.58	0.41	0.79	0.96	0.91	0.72	0.74	0.67	0.78	0.92	0.88	0.00	0.01	-0.40	-0.91	0.05	0.06	0.15	0.40	1.69	0.32	0.41
NWJ	0.22	0.40	0.21	0.34	0.41	0.42	0.41	0.85	1.00	0.97	1.02	0.29	0.91	1.26	1.01	0.41	0.98	0.58	1.55	0.41	0.80	0.96	0.91	0.72	0.74	0.67	0.79	0.93	0.88	0.01	0.00	0.03	-0.78	0.06	0.06	0.12	0.40	1.68	0.32	0.41
Classic estimators																																									
KSG	0.23	0.38	0.18	0.24	0.41	0.41	0.41	0.17	0.89	0.65	1.04	0.30	0.19	1.07	0.96	0.42	0.74	0.57	1.29	0.41	0.20	0.92	0.71	0.17	0.72	0.55	0.19	0.90	0.69	0.17	0.22	0.09	0.12	0.07	0.20	0.15	0.42	1.68	0.32	0.41
LNN	0.25	0.89	2.72	6.66	0.41	0.42	0.42	17.34	2.68	6.45	1.27	0.65	17.34	17.34	3.10	2.48	7.31	6.77	33.48	0.39	17.34	2.49	7.27	17.34	3.10	7.31	17.34	2.38	7.24	-0.31	-0.70	0.49	1.01	2.11	2.47	5.49	0.34	1.48	0.29	0.42
CCA	0.00	0.00	0.00	0.00	0.34	0.41	0.38	0.98	0.97	0.95	1.03	0.30	1.04	1.32	1.03	0.42	1.02	0.59	1.75	0.39	0.99	0.96	0.96	0.85	0.23	0.39	0.98	0.84	0.89	0.42	2.15	0.12	0.39	0.01	0.48	0.02	0.02	1.63	0.18	0.38

G SINGLE-CELL SIGNALING RESPONSES: TECHNICAL DETAILS

This section complements Section 4.1: it provides additional details and results. The reference study for this section is Jetka et al. (2019), for which there are no ground truth MI estimates: as such, throughout this section, we compare against the biological conclusions of that study, and note that MI estimates are essentially equivalent to our results.

Application domain. Cells sense extracellular cues through signaling pathways that convert ligand concentrations into effector activity and gene regulation. A canonical example is the NF-$\mathcal{K}$B pathway, which responds to the inflammatory cytokine TNF-α and regulates immune responses; although the underlying biochemistry is well characterized, how reliably individual cells infer stimulus strength from their response trajectories remains unclear (Purvis & Lahav, 2013; Lee et al., 2014; Antebi et al., 2017). Over the past two decades, cellular signaling has increasingly been formulated in terms of information theory (Nurse, 2008; Waltermann & Klipp, 2011; Brennan et al., 2012; Jetka et al., 2018; Petkova et al., 2019): an extracellular stimulus (X) is transmitted through a stochastic biochemical network to produce a cellular response (Y), so mutual information $\mathrm{I}(X;Y)$ quantifies how much observing the response reduces uncertainty about the stimulus, while channel capacity measures the maximum information transmissible over input distributions. This perspective has enabled measurements of signaling fidelity in pathways such as TNF-α–NF-$\mathcal{K}$B and has shown that time-resolved response trajectories can transmit more information than static measurements (Tostevin & Ten Wolde, 2009; Cheong et al., 2011; Selimkhanov et al., 2014).

Estimand and metrics. The input X is an experimentally controlled stimulus taking one of m values with input distribution $p(X)$, and the output $Y \in \mathbb{R}^d$ is a vector of single-cell measurements distributed according to the unknown conditionals $P(Y \mid X = x_i)$. Mutual information decomposes as $\mathrm{I}(X;Y) = \sum_i p_i D_i$, where $D_i = \mathrm{KL}\left[P(Y \mid x_i) \,\|\, \bar{P}\right]$ is the divergence of each dose-conditional from the output mixture $\bar{P} = \sum_j p_j P(Y \mid x_j)$. We estimate each D_i with the conditional variant of the estimand (Appendix B.4). Capacity $C = \max_p \mathrm{I}(X;Y)$ is computed by the Blahut–Arimoto algorithm (Blahut, 1972; Arimoto, 1972) run directly on the estimated per-dose divergences: since the conditional fields do not depend on p, they are cached once, and only the mixture field is re-estimated as the ascent updates $p_i \propto p_i e^{D_i}$. At each iteration the shuffled context of Appendix B.4 is redrawn with the current p: doses drawn from p select the pool from which each response row is taken, and the dose column is overwritten by an independent draw from p, so the response marginal of the context is $\sum_i p_i P(Y \mid x_i)$ and every D_i is measured against the mixture of the current iterate. The sampling noise of the redraw is held fixed across iterations by reseeding from one base seed, so the ascent is a deterministic function of p. For the pairwise probability of correct discrimination (PCD), which is the Bayes accuracy of deciding between doses i and j from a single cell under equal priors, we exploit the fact that the two-dose mutual information at $p = (\frac{1}{2}, \frac{1}{2})$ equals the Jensen–Shannon divergence J_{ij}, which brackets the Bayes accuracy as $\frac{1}{2}(1 + J_{ij}) \le \mathrm{PCD}_{ij} \le \frac{1}{2}\big(1 + \min(1, \sqrt{2 \ln 2 \cdot J_{ij}})\big)$ (with J in bits).

Results in full. Figure 10 shows the six-panel version of Figure 4, with the pairwise discrimination matrices. The precise numbers presented in Section 4.1 are the following. The single-frame capacity (panel B) peaks at 1.12 to 1.16 bits at minutes 15 to 21, against about 1 bit in Jetka et al. (2019); it falls to 0.04 bits at minute 66, and the second rise reaches 0.40 bits at minute 93. The prefix capacity is 0.87 bits with the first three frames, 1.29 with the first five, 1.34 with the first nine, and between 1.07 and 1.34 afterwards, so the prefix of frames exceeds the best single frame. The trajectory capacity is $C \approx 1.05$ bits (0.93 to 1.20 across seeds), against 1.3 bits in the reference study. The PCD averages 0.74 over the 55 dose pairs for the single frame at minute 21 and 0.84 for the trajectory; over the 15 pairs of doses at or above 0.5 ng/ml, where the amplitude of the first peak saturates, the averages are 0.56 and 0.67, so the gain from dynamics is concentrated at high doses.

H PROMOTER IDENTIFICATION: TECHNICAL DETAILS

This section expands on the application domain, the dataset, the encoding, and the diagnostics behind the results of Section 4.2, and reports the numbers that the main text summarizes.

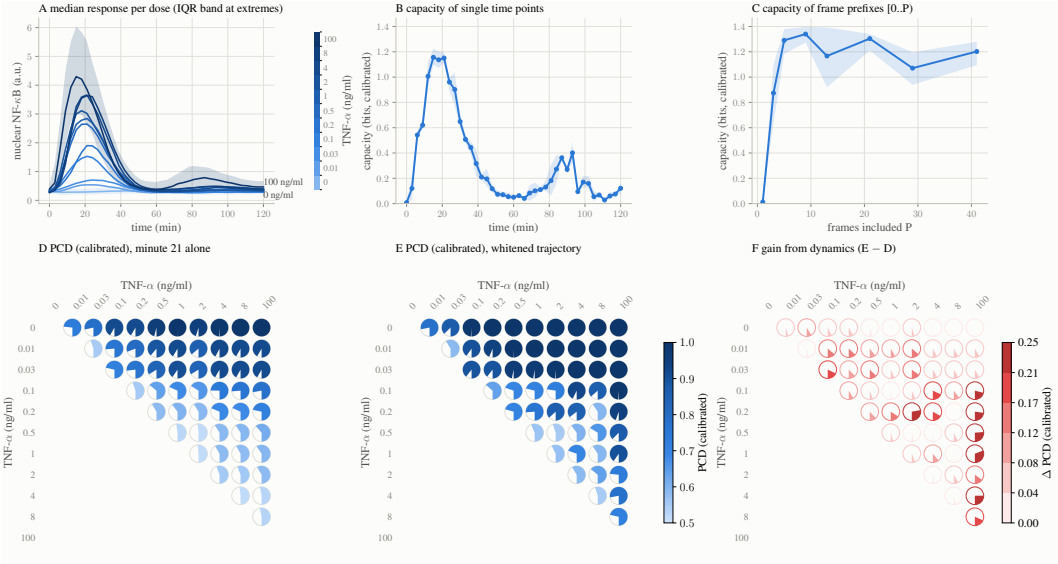


Figure 10: Six-panel analysis of the NF-$\mathcal{K}$B dose channel. **(A–C)** As in Figure 4. **(D–F)** Pairwise probability of correct discrimination between doses: from the single frame at minute 21 (D), from the trajectory (E), and the gain from dynamics (F, E minus D). The filled fraction of each circle and its color both encode the value, from chance (0.5) to certain discrimination (1) in D and E, and from 0 to 0.25 in F.

Application domain. Genomics relies on computational methods to find patterns in large datasets from basic and clinical research (Libbrecht & Noble, 2015; Whalen et al., 2022; Teschendorff & Horvath, 2025). DNA is a sequence of four bases, adenine (A), thymine (T), guanine (G), and cytosine (C), and the order of the bases determines the biological instructions that a strand of DNA carries. We follow the recent practice of treating DNA sequences as text (Dotan et al., 2024; Qiao et al., 2024; Malusare et al., 2024; Eapen, 2025), with the simplest tokenization: each base is one token, so a sequence is a high-dimensional vector whose coordinates take four values. A central question in molecular biology is the regulation of gene expression: expression requires a stretch of regulatory DNA called a promoter, which contains motifs, that is, patterns whose presence shows a statistically significant dependence with expression levels. Computational methods based on MI (Elemento et al., 2007; Rao et al., 2007) search whole genomes for the key elements of transcription regulation by quantifying the dependence between the presence of a motif in a regulatory region and the expression of the corresponding gene; further motif properties, such as position bias, orientation preference, and functional interactions, can be studied with MI as well (Elemento et al., 2007). A minimal eukaryotic promoter contains a transcription start site (TSS) and a TATA-BOX motif about 30 base pairs upstream of the TSS; in *Arabidopsis thaliana* the preferred position is between -39 and -26 relative to the TSS (Bernard et al., 2010).

Dataset. Umarov & Solovyev (2017) evaluate convolutional promoter-recognition models on sequences extracted from the EPD database (Dreos et al., 2013). We use their *Arabidopsis thaliana* TATA-promoter and non-promoter collection, 1,497 and 2,879 sequences respectively, each of 251 bases; promoter sequences span positions -200 to $+50$ around the annotated TSS. We discard sequences containing ambiguous bases and subsample the non-promoter class to 1,497 sequences, so the promoter label X is uniform and the MI of every window is bounded by $H(X) = \ln 2$ nats.

Relation to previous work on motif search. Umarov & Solovyev (2017) localize functional elements by substituting a sliding region of the input with random bases and tracking the drop in classification accuracy. Foresti et al. (2026) uses a recent MI estimator based on discrete diffusion to reproduce the same protocol of Umarov & Solovyev (2017). The MI profile we obtain in Section 4.2 recasts this search using ALICE: windows on segments irrelevant to promoter status yield values near zero, and windows overlapping the TATA-BOX motif yield high values.

Encoding. The velocity fields of Equation (3) are defined on $\mathbb{R}^d$, so discrete symbols are transformed by a fixed injective embedding: each symbol maps to a vector of K real coordinates, where every coordinate holds an independent random permutation of equally spaced standard-normal quantiles, perturbed by a small uniform dither confined within each level. Injectivity preserves $\mathrm{I}(X;Y)$ exactly for every K; the dither removes the ties that would otherwise collapse the Gaussian-copula normalization of the estimator, and each encoded coordinate is approximately standard normal, the scale on which ALICE is trained. A window of L bases concatenates its per-base vectors, giving blocks of dimension K for the label and LK for the window; the unequal, length-dependent widths are handled natively by the variable-dimension capabilities of ALICE. The reported results use $K = 1$, the minimal injective width.

Protocol and numbers. For each of the 246 ($L = 6$) or 248 ($L = 4$) window positions, ALICE conditions on a context of 1,024 encoded label–window pairs, and the velocity differences are averaged over 1,024 held-out pairs and 32 time points. Both window lengths place the maximum at 30 to 32 bases upstream of the TSS, inside the documented TATA-BOX band: the peak is 0.31 nats at offset -30 for $L = 4$ over 248 windows, and 0.34 nats at offset -32 for $L = 6$ over 246 windows.

Discussion. Both window lengths place the top windows at TSS offsets -32 to -30, and both resolve the two core promoter elements the sequences carry: the TATA-BOX at -30, whose top 6-mers (TATATA, TATAAA) each occur in about 4% of promoters against about 0.4% for the most frequent non-promoter 6-mer, and the initiator element straddling the TSS, which is a pyrimidine/purine pair (position -1 is C or T in 94% of promoters, position $+1$ is A or G in 93%) and has no recurring k-mer. Since the promoter set is the TATA-containing subset of EPD, the -30 peak acts as a positive control for localization.

I BRAIN REGION ACTIVITY PATTERNS: DETAILS

This Section provides additional details about the Ω-info estimator used in Section 4.3, the structure of the Visual Behavior Neuropixels data, the selection of flashes that produced the analyzed tables, the experimental protocol, and the full per-window results.

Estimator. For N blocks $X = (X_1, \ldots, X_N)$ the total correlation and the dual total correlation are $\mathrm{TC} = \mathrm{KL}\big(p(x) \,\|\, \prod_i p(x_i)\big)$ and $\mathrm{DTC} = H(X) - \sum_i H(X_i \mid X_{\setminus i})$, where $X_{\setminus i}$ denotes all blocks except X_i, and the Ω-info is $\Omega = \mathrm{TC} - \mathrm{DTC}$ (Rosas et al., 2019). Bounoua et al. (2024) write both terms as time integrals of squared score differences, evaluated at the same noised coordinates: for TC, between the joint score and the concatenation of the N marginal scores; for DTC, between the joint score and the concatenation of the N scores of each block conditioned on the clean values of the other blocks. Under the interpolant $x_t = (1-t)x_0 + t\,\varepsilon$ of Section 2, the score of a block and its velocity are related by $s = ((1-t)v - x_t)/t$, so two fields that share the noised coordinate differ by $\Delta s = \frac{1-t}{t}\Delta v$. With the weight of Equation (4) this gives

$$\mathrm{TC} = \int_0^1 \frac{1-t}{t}\, \mathbb{E} \sum_{i=1}^{N} \big\| v_\theta(x_t, t; C)\big|_i - v_\theta\big(x_{i,t}, t; C_i\big) \big\|^2 \, \mathrm{d}t, \tag{14}$$

$$\mathrm{DTC} = \int_0^1 \frac{1-t}{t}\, \mathbb{E} \sum_{i=1}^{N} \big\| v_\theta(x_t, t; C)\big|_i - v_\theta\big([x_{i,t}, x_{0,\setminus i}], t, \mathbb{1}_i; C\big)\big|_i \big\|^2 \, \mathrm{d}t, \tag{15}$$

where C_i is the context restricted to the columns of block i, $\mathbb{1}_i$ is the noising indicator that noises block i and holds the other blocks at their clean values, and $|_i$ selects the coordinates of block i. The marginal field in Equation (14) requires no dedicated mechanism: ALICE accepts any joint width, so the field conditioned on C_i is the field of the marginal law of X_i. The conditional field in Equation (15) is the masked field of Equation (4), and for $N = 2$ Equation (15) is the MI estimator of Section 2. Equation (15) rests on the identity $\mathbb{E}\big[s_{i|\setminus i}(x_{i,t}; x_{0,\setminus i}) \,\big|\, x_t\big] = s(x_t)|_i$. Given the clean values of the other blocks, the noised block i and the noised other blocks are independent, so the conditional score of block i equals the block-i score of $p_t(x_t \mid x_{0,\setminus i})$; averaging that score over $p(x_{0,\setminus i} \mid x_t)$ gives the joint score. All fields are evaluated under common random numbers: the same (x_0, t, ε) draw is used for every term. Per Monte-Carlo row, the estimator evaluates $2N + 1$ velocities:

one joint, N conditional, and N marginal. Both integrals are invariant under any per-block bijection, so the copula normalization of Section 2, which is applied per coordinate, leaves them unchanged, and the sub-context fields see the normalized columns of the joint context.

Task and trial structure. One image is shown to a mouse for 250 ms followed by 500 ms of gray screen, so a new flash starts every 750 ms. The image repeats over several flashes and then changes; the mouse earns water by licking after a change. A trial is one run of repeats together with the change flash that ends it, and the next trial repeats the image that the change introduced. The position of a flash is its rank inside its trial, counting from one. About 5% of flashes are omitted by design (the screen stays gray). The active block of a session holds about 4,800 flashes. Neuropixels probes record single units in the areas VISp, VISl, VISal, VISrl, VISam, and VISpm. We use the 72 sessions of Bounoua et al. (2024): mice with a familiar-image and a novel-image session, recorded on consecutive days, with more than 20 well-isolated units (signal-to-noise ratio above 1 and fewer than one inter-spike-interval violation) in each of the six areas. The pairing of the two sessions of a mouse and their order (familiar first) were verified against the session table of the Allen Institute.

Selection of flashes. The preprocessing of Bounoua et al. (2024) is designed to keep, for both flash types, only trials in which the mouse was rewarded, and to drop non-change flashes during which the mouse licked. We reproduced their tables exactly from the raw spike times (identical row counts and values in every session we compared) and found that neither filter has an effect in the released code: the reward filter tests a field that is always missing and is therefore always satisfied, and the lick exclusion is negated twice and is also always satisfied. The analyzed tables are therefore defined as follows. A change flash is any flash of the active block at which the image changed. A non-change flash is any non-omitted flash of the active block at positions 4 to 10 of its trial that still shows the image the trial started with. A session holds 145 to 351 change flashes and 1,154 to 1,715 non-change flashes. On the first session, for example, the 253 change flashes comprise 203 hits and 47 misses, and the 1,476 non-change flashes comprise 518 flashes of hit trials, 154 of miss trials, 546 of aborted trials, and 247 of catch trials, 111 of them with a lick. We keep this selection so that our estimates and those of Bounoua et al. (2024) describe the same rows; the outcome of every trial is stored with every flash, so the hit-only and lick-free selections need no new data. Positions 1 to 3 of a trial are excluded because the response to a repeated image decreases over the first repeats and levels off from the fourth.

Windows, step size, and dimension. For every flash and unit, spikes are counted in 250 bins of 1 ms after flash onset, averaged over the units of an area, cut into five windows of 50 ms, and summed inside each window in steps of s ms. The dimension of the variable of one area in one window is therefore $50/s$: one number at $s = 50$ ms (used in Section 4.3), 25 numbers at $s = 2$ ms (the main figure of Bounoua et al. (2024)), and 10 or 50 at $s = 5$ or 1 ms (their appendix). The joint width is the number of areas times this dimension. The file distributed with Bounoua et al. (2024) contains the tables at $s = 50$ ms; the finer resolutions were rebuilt from the raw spike times. The rebuilt 50 ms tables reproduce the distributed ones exactly: estimating the 3,375 per-session values from the rebuilt tables returns the same numbers to machine precision.

Correlation between flashes. Rows of a session are flashes in temporal order, and consecutive flashes are correlated. Table 5 reports the lag autocorrelation of the six-area vector within a session at 100 to 150 ms. Change flashes occur once per trial and are close to independent; non-change flashes occur about six times per trial, 750 ms apart, and are strongly correlated at short lags. The effective number of independent draws per session, from the truncated autocorrelation sum, is 185 for change flashes (of 251 nominal rows, median over sessions) and 146 to 205 for non-change flashes (of 1,447). The two flash types therefore carry a similar amount of information despite a six-fold difference in row count. Two consequences follow for the protocol. Sample sizes are quoted as effective draws. The context and the evaluation rows of a session are split by contiguous runs, because a random split places repeats of one trial on both sides, and the evaluation points then have near copies in the context.

Protocol details. Each session, window, and flash type is estimated from the rows of that session with ALICE-Base: we use a context of 128 rows and an evaluation set of the remaining rows, capped at 512, assigned by contiguous runs of 32 rows, 64 time draws per evaluation row, and five context

Table 5: Within-session autocorrelation of the six-area vector at lag k (in flashes), mean over areas and sessions, 100 to 150 ms window.

k	1	2	3	5	10	20	50	100
change	-0.02	$+0.07$	$+0.07$	$+0.07$	$+0.06$	$+0.05$	$+0.01$	-0.03
non-change	$+0.47$	$+0.38$	$+0.33$	$+0.25$	$+0.15$	$+0.10$	$+0.08$	$+0.07$

Table 6: Median over sessions of the per-session Ω-info (nats) of the six areas, context of 128 rows drawn from the same session, by window after flash onset. Interquartile range in brackets.

sessions	flash type	0–50	50–100	100–150	150–200	200–250 ms
familiar ($n = 31$)	change	0.34 [0.25, 0.47]	0.24 [0.18, 0.32]	0.41 [0.29, 0.59]	0.38 [0.24, 0.56]	0.22 [0.19, 0.33]
familiar ($n = 36$)	non-change	0.34 [0.22, 0.48]	0.25 [0.18, 0.37]	0.41 [0.25, 0.51]	0.69 [0.50, 0.85]	0.35 [0.28, 0.53]
novel ($n = 32$)	change	0.16 [0.12, 0.24]	0.83 [0.69, 0.96]	1.08 [0.88, 1.27]	0.66 [0.51, 0.79]	0.37 [0.24, 0.47]
novel ($n = 36$)	non-change	0.17 [0.11, 0.26]	0.55 [0.45, 0.73]	0.60 [0.48, 0.75]	0.49 [0.39, 0.73]	0.28 [0.22, 0.41]

draws. The context size is the largest one that keeps most change sessions: 192 rows are required for 128 context rows and 64 evaluation rows, and sessions with fewer change flashes are excluded from the change condition, which leaves 31 familiar and 32 novel sessions for change flashes and all 36 of each kind for non-change flashes. The same context size is applied to both flash types because the estimate depends on the context size. In a first run with the context of each flash type set by its own row count (128 to 150 rows for change flashes and 1,024 for non-change flashes), the two flash types differed already in the first window, before the visual response ($+0.12$ nats, $p = 2 \cdot 10^{-8}$ over 63 sessions); with matched contexts the first-window difference is $+0.01$ nats ($p = 0.9$) and the peak difference is unchanged. Two further controls quantify the choices above. Assigning rows to the context at random, without the run structure, changes the estimates by 0.01 nats at this dimension. Drawing the 128 context rows from the other 71 sessions raises the median estimate at the peak from 0.49 to 0.75 nats for non-change flashes and from 0.70 to 0.83 for change flashes, and removes most of the difference between familiar- and novel-image sessions (medians of 0.80 and 0.87 nats for change flashes, against 0.41 and 1.08 with own-session contexts); a context pooled across animals describes a mixture whose components share the animal-specific level of activity, and that shared component is attributed to redundancy. Between-session differences account for 27 to 56% of the variance of every column of the pooled table.

Per-window results. Figure 11 completes Figure 6 with the familiar-image sessions, the within-session difference between the two flash types for both image sets, and the difference between the two sessions of a mouse for non-change flashes. Table 6 lists the median over sessions of the per-session estimates behind both figures, and Table 7 the paired contrasts with signed-rank tests. In novel-image sessions the Ω-info is positive in every window for every mouse, and the maximum over windows falls at 100 to 150 ms for 78% of the mice for change flashes and for 47% for non-change flashes. In familiar-image sessions the change-flash profile is flat, and the non-change profile rises late, with its maximum at 150 to 200 ms. The familiar session of a mouse precedes the novel one by one day, and the two sessions already differ before the visual response arrives, by -0.16 nats in the first window. Table 8 decomposes the variance of the per-session estimates by sequential sums of squares of the crossed factors; the seed share is the variability of the estimator across context draws. Across all 63 sessions with both flash types, animal identity accounts for 38% of the variance of the peak-window estimates, experience level for 23%, flash type for 8%, and the variability of the estimator across context draws for 10%.

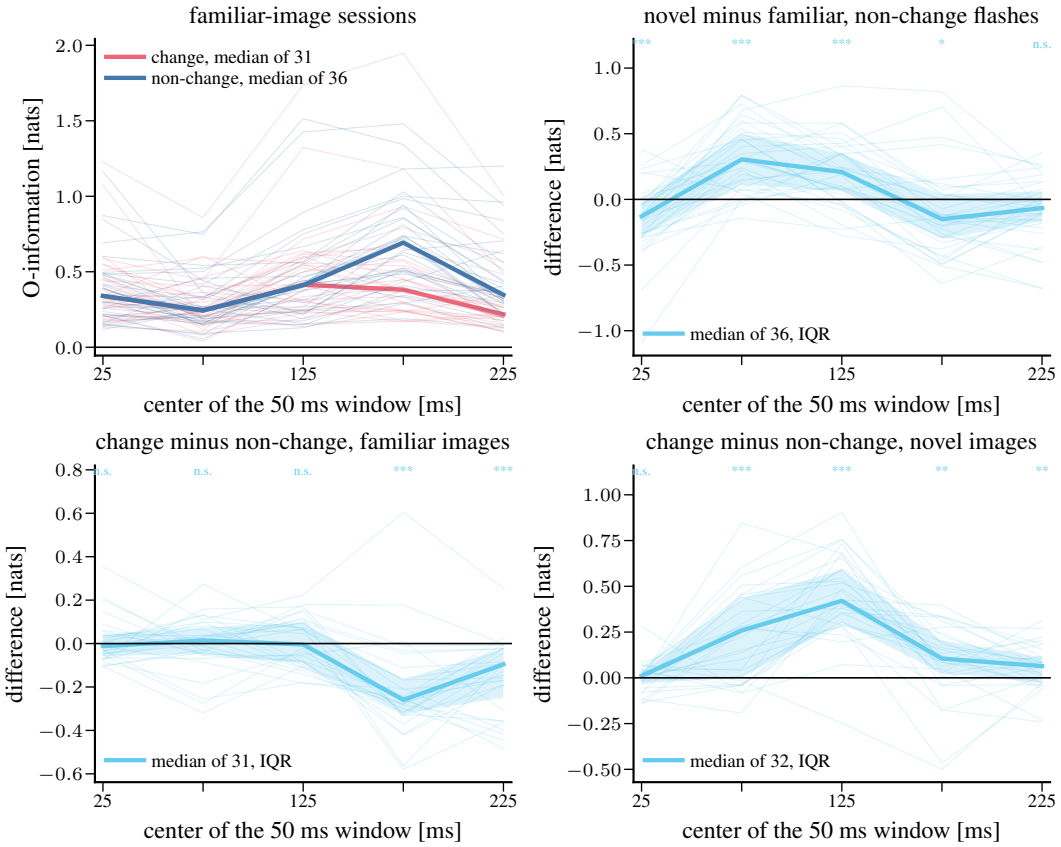


Figure 11: Complement of Figure 6: Ω-info of the six visual areas in the five 50 ms windows after a flash, one estimate per session. Top left: familiar-image sessions, one thin line per mouse and flash type, group medians in bold. Top right: difference between the novel and the familiar session of each mouse for non-change flashes. Bottom: difference between change and non-change flashes within each session, for familiar-image (left) and novel-image (right) sessions. Difference panels show the median, the interquartile band, and a signed-rank test per window ($*$: $p < 0.05$, $**$: $p < 0.01$, $***$: $p < 0.001$). The plotted values are stored with the figure.

Table 7: Paired contrasts of the per-session estimates (nats): median difference, two-sided Wilcoxon signed-rank p-value, and fraction of positive differences, by window.

contrast	group	0–50	50–100	100–150	150–200	200–250 ms
change − non-change, within session	familiar ($n = 31$)	-0.01 (0.8, 48%)	$+0.01$ (0.8, 52%)	-0.00 (0.8, 48%)	-0.26 ($8 \cdot 10^{-6}$, 10%)	-0.10 ($2 \cdot 10^{-6}$, 6%)
change − non-change, within session	novel ($n = 32$)	$+0.01$ (0.9, 56%)	$+0.26$ ($3 \cdot 10^{-5}$, 75%)	$+0.42$ ($3 \cdot 10^{-9}$, 97%)	$+0.10$ ($3 \cdot 10^{-3}$, 81%)	$+0.06$ ($5 \cdot 10^{-3}$, 72%)
novel − familiar, within mouse	change ($n = 27$)	-0.16 ($8 \cdot 10^{-4}$, 15%)	$+0.51$ ($3 \cdot 10^{-8}$, 96%)	$+0.56$ ($7 \cdot 10^{-8}$, 93%)	$+0.22$ ($7 \cdot 10^{-4}$, 81%)	$+0.11$ ($1 \cdot 10^{-3}$, 81%)
novel − familiar, within mouse	non-change ($n = 36$)	-0.13 ($8 \cdot 10^{-4}$, 22%)	$+0.30$ ($2 \cdot 10^{-8}$, 86%)	$+0.21$ ($2 \cdot 10^{-5}$, 78%)	-0.15 (0.01, 31%)	-0.07 (0.09, 33%)

Table 8: Variance shares of the per-session estimates (all 63 sessions with both flash types, five context draws each) by window: sequential sums of squares of mouse identity, experience level, and flash type; the seed share is the variability of the estimator across context draws; the remainder holds interactions.

window (ms)	mouse	experience	flash type	seed	remainder
0–50	41%	10%	0%	20%	29%
50–100	21%	40%	3%	13%	24%
100–150	38%	23%	8%	10%	22%
150–200	54%	0%	1%	14%	31%
200–250	55%	0%	1%	14%	29%