

Article

The Information Bottleneck's Ordinary Differential Equation: First-Order Root Tracking for the Information Bottleneck

Shlomi Agmon 

School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem 9190401, Israel; shlomi.agmon@mail.huji.ac.il

Abstract: The Information Bottleneck (IB) is a method of lossy compression of relevant information. Its rate-distortion (RD) curve describes the fundamental tradeoff between input compression and the preservation of relevant information embedded in the input. However, it conceals the underlying dynamics of optimal input encodings. We argue that these typically follow a piecewise smooth trajectory when input information is being compressed, as recently shown in RD. These smooth dynamics are interrupted when an optimal encoding changes qualitatively, at a *bifurcation*. By leveraging the IB's intimate relations with RD, we provide substantial insights into its solution structure, highlighting caveats in its finite-dimensional treatments. Sub-optimal solutions are seen to collide or exchange optimality at its bifurcations. Despite the acceptance of the IB and its applications, there are surprisingly few techniques to solve it numerically, even for finite problems whose distribution is known. We derive anew the IB's first-order Ordinary Differential Equation, which describes the dynamics underlying its optimal tradeoff curve. To exploit these dynamics, we not only detect IB bifurcations but also identify their type in order to handle them accordingly. Rather than approaching the IB's optimal tradeoff curve from sub-optimal directions, the latter allows us to follow a solution's trajectory along the optimal curve under mild assumptions. We thereby translate an understanding of IB bifurcations into a surprisingly accurate numerical algorithm.

Keywords: Information Bottleneck; bifurcations; ordinary differential equation; numerical approximation



Citation: Agmon, S. The Information Bottleneck's Ordinary Differential Equation: First-Order Root Tracking for the Information Bottleneck. *Entropy* **2023**, *25*, 1370. <https://doi.org/10.3390/e25101370>

Academic Editors: Jan Lewandowski and Gerhard Bauch

Received: 16 May 2023

Revised: 8 August 2023

Accepted: 9 August 2023

Published: 22 September 2023



Copyright: © 2023 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Information Bottleneck (IB) describes the fundamental tradeoff between the compression of information on an *input* X to the preservation of relevant information on a hidden *reference* variable Y . Formally, let X and Y be random variables defined, respectively, on finite *source* and *label alphabets* \mathcal{X} and \mathcal{Y} , and let $p_{Y|X}(y|x)p_X(x)$ be their joint probability distribution, or $p(y|x)p(x)$ for short (without loss of generality, $p(x) > 0$ for every $x \in \mathcal{X}$ and so $p_{Y|X}$ is well-defined). One seeks [1] to maximize the information $I(Y; \hat{X})$ over all Markov chains $Y \longleftrightarrow X \longleftrightarrow \hat{X}$, subject to a constraint on the mutual information $I(X; \hat{X}) := \mathbb{E}_{p(\hat{x}|x)p(x)} \log \frac{p(\hat{x}|x)}{p(\hat{x})}$,

$$I_Y(I_X) := \max_{p(\hat{x}|x)} \{I(Y; \hat{X}) : I(X; \hat{X}) \leq I_X\}. \quad (1)$$

The latter maximization is over conditional probability distributions or *encoders* $p(\hat{x}|x)$. The graph of $I_Y(I_X)$ is the *IB curve*. We write $T := |\hat{\mathcal{X}}|$, for a codebook or *representation alphabet* $\hat{\mathcal{X}}$. An encoder $p(\hat{x}|x)$ which achieves the maximum in (1) is *IB optimal* or simply *optimal*.

Written in a Lagrangian formulation $\mathcal{L} := I(X; \hat{X}) - \beta I(Y; \hat{X})$ with $\beta > 0$ (normalization constraints omitted for clarity), [1] showed that a necessary condition for extrema in (1) is that the *IB Equations* hold. Namely,

$$p(\hat{x}|x) = \frac{p(\hat{x})}{Z(x, \beta)} \exp \{ -\beta D_{KL}[p(y|x)||p(y|\hat{x})] \} , \tag{2}$$

$$p(y|\hat{x}) = \sum_x p(y|x)p(x|\hat{x}) , \quad \text{and} \tag{3}$$

$$p(\hat{x}) = \sum_x p(\hat{x}|x)p(x) . \tag{4}$$

In these, $Z(x, \beta) := \sum_{\hat{x}} p(\hat{x}) \exp \{ -\beta D_{KL}[p(y|x)||p(y|\hat{x})] \}$ is the *partition function*, D_{KL} is the Kullback–Leibler divergence, $D_{KL}[p||q] := \sum_i p(i) \log p^{(i)}/q^{(i)}$, and $p(x|\hat{x})$ in (3) is defined by the Bayes rule $p(\hat{x}|x)p(x)/p(\hat{x})$. The IB Equations (2)–(4) are a necessary condition for an extremum of \mathcal{L} also when it is considered as a functional in three independent families of normalized distributions $\{p(\hat{x}|x)\}$, $\{p(y|\hat{x})\}$ and $\{p(\hat{x})\}$, ref. [1] (Section 3.3), rather than in $\{p(\hat{x}|x)\}$ alone. While satisfying them is necessary to achieve the curve (1), it is not sufficient. Indeed, Equations (2)–(4) have solutions that do not achieve curve (1), and so are *sub-optimal*. This results in sub-optimal IB curves, which intersect or *bifurcate* as the multiplier β varies (see Section 3.4 in [1]).

Iterating over the IB Equations (2)–(4) is essentially Blahut–Arimoto’s algorithm variant for the IB (BA-IB) due to [1], brought below for reference. While the minimization problem (1) can be solved exactly in special cases [2] (Section IV), exact solutions of an arbitrary finite IB problem whose distribution is known are usually obtained nowadays using BA-IB; see [3] (Section 3) for a survey of other computation approaches. Write BA_β for a single iteration of BA-IB. Since BA_β encodes an iteration over the IB Equations (2)–(4), then an encoder $p(\hat{x}|x)$ is its fixed point, $BA_\beta[p(\hat{x}|x)] = p(\hat{x}|x)$, if and only if it satisfies the IB Equations. Or equivalently, if $p(\hat{x}|x)$ is a root of the *IB operator*,

$$F := Id - BA_\beta , \tag{5}$$

in a manner similar to [4]. We shall then call it an *IB root*. Agmon et al. [4] used a similar formulation of rate-distortion (RD) and its relations in [5] to the IB, to show that BA-IB suffers from *critical slowing down* near *critical points*, where the marginal $p(\hat{x})$ of a representor \hat{x} in an optimal encoder vanishes gradually. That is, the number of BA-IB iterations required until convergence increases dramatically as one approaches such points.

Formulating fixed points of an iterative algorithm as operator roots can also be leveraged for computational purposes in a constrained optimization problem, as noted recently by [6] for RD. Indeed, let $F(\cdot, \beta)$ be a differentiable operator on \mathbb{R}^n for some $n > 0$, $F : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$, where β is a (real) constraint parameter. Suppose now that (x, β) is a root of F ,

$$F(x, \beta) = \mathbf{0} , \tag{6}$$

such that $x = x(\beta)$ is a differentiable function of β . Write $D_x F := (\frac{\partial}{\partial x_j} F_i)_{i,j}$ for its Jacobian matrix, and $D_\beta F := (\frac{\partial}{\partial \beta} F_i)_i$ for its vector of partial derivatives with respect to β . The point (x, β) of evaluation is omitted whenever understood. As is often discussed along with the Implicit Function Theorem, e.g., [7], applying the multivariate chain rule to $F(x(\beta), \beta)$ in (6) yields an implicit ordinary differential equation (ODE)

$$D_x F \frac{dx}{d\beta} = -D_\beta F , \tag{7}$$

for the roots of F . Plugging in explicit expressions for the first-order derivative tensors $D_x F$ and $D_\beta F$, one can specialize (7) to a particular setting, which allows one to compute the *implicit derivatives* $\frac{dx}{d\beta}$ numerically. While [6] discovered the *RD ODE* this way, they showed that (7) can be generalized to arbitrary order under suitable differentiability assumptions. Namely, they showed that the derivatives $\frac{d^l x}{d\beta^l}$ implied by $F = \mathbf{0}$ (6) can be computed via a recursive formula, for an arbitrary-order $l > 0$. By specializing this with the higher

derivatives of Blahut's algorithm [8], they obtained a family of numerical algorithms for following the path of an optimal RD root (Part I there).

In this work, we specialize the implicit ODE (7) to the IB. Namely, we plug into (7) the first-order derivatives of the IB operator $Id - BA_\beta$ (5) to obtain the IB ODE, and then use it to reconstruct the path of an optimal IB root, in a manner similar to [6]. This is not to be confused with the gradient flow (of arbitrary encoders) towards an optimal root at a fixed β value, described in [9] (Equation (6)) by an ODE, which is a different optimization approach. In contrast, the implicit Equation (7) describes how a root evolves *with* β . So, in principle, one may compute an optimal IB root once and then follow its evolution along the IB curve (1). While the discovery of the IB ODE is due to [10], we derive it here anew in a form that is better suited for computational (and other) purposes, especially when there are fewer possible labels \mathcal{Y} than input symbols \mathcal{X} , as often is the case. To that end, we consider several natural choices of a coordinate system for the IB in Section 2 and compare their properties. This allows us to make an apt choice for the ODE's variable x in (7). In Section 3, we present the IB ODE in these coordinates (Theorem 1). This enables one to numerically compute the first-order implicit derivatives at an IB root, if it can be written as a differentiable function in β . So long as an optimal root remains differentiable, a simple way to reconstruct its trajectory is by taking small steps at a direction determined by the IB ODE. This is *Euler's method* for the IB. The error accumulated by Euler's method from the true solution path is roughly proportional to the step size, when small enough. For comparison, reverse deterministic annealing [11] with BA-IB is nowadays common for computing IB roots. The dependence of its error on the step size is roughly the same as in Euler's method. This is discussed in Section 4, where we combine Euler's method with BA-IB to obtain a modified numerical method whose error decreases at a faster rate than either of the above.

However, the differentiability of optimal IB roots breaks where the solution changes qualitatively. Such a point is often called a phase transition in the IB literature, or a *bifurcation*—namely, a point where there is a change in the problem's number of solutions; e.g., [12] (Section 2.3) for basic definitions. As noted already by Tishby et al. in [1], their existence in the IB stems from restricting the cardinality of the representation alphabet $\hat{\mathcal{X}}$. Since IB roots are the solutions of the fixed-point Equations (2)–(4), then the gap between achieving the IB curve (1) to merely satisfying these Equations lies in understanding the solution structure of the IB operator (5), or equivalently its bifurcations. While IB bifurcations were analyzed in several works, including [9,13,14] and others, little is known about the practical value of understanding them. In [15,16] it was shown that they correspond to the onset of learning new classes, and in [4] that they inflict a hefty computational cost to BA-IB. Following [6], this work demonstrates that understanding bifurcations can be translated to a new numerical algorithm to solve the IB. To that end, merely detecting a bifurcation along a root's path does not suffice. Rather, it is also necessary to identify its type, as this allows one to handle the bifurcation accordingly. One can then continue following the path dictated by the IB ODE.

Almost all of the literature on IB bifurcations is based on a perturbative approach, in a manner similar to [17] (Section IV.C). That is, suppose that the first variation

$$\left. \frac{\partial}{\partial \epsilon} \mathcal{L}[p(\hat{x}|x) + \epsilon \Delta p(\hat{x}|x); \beta] \right|_{\epsilon=0} \quad (8)$$

of the IB Lagrangian \mathcal{L} vanishes, for every perturbation $\Delta p(\hat{x}|x)$. This condition is necessary for extremality and implies [1] the IB Equations (2)–(4). Then, $(p(\hat{x}|x), \beta)$ is said to be a *phase transition* only if there exists a particular direction $\Delta q(\hat{x}|x)$ at which $p(\hat{x}|x)$ can be perturbed without affecting the Lagrangian's value to second order,

$$\left. \frac{\partial^2}{\partial \epsilon^2} \mathcal{L}[p(\hat{x}|x) + \epsilon \Delta q(\hat{x}|x); \beta] \right|_{\epsilon=0} = 0. \quad (9)$$

For finite IB problems, condition (8) boils down to requiring that the gradient of \mathcal{L} vanishes, while condition (9) is equivalent to requiring that its Hessian matrix has a non-trivial kernel (as both are conditions on the directional derivatives, e.g., [18]). The works [9,14–16] take such an approach, while [13] focuses on one type of IB bifurcations.

While a perturbative approach is common in analyzing phase transitions, it has several shortcomings when applied to the IB, as noted by [10]. First, the IB's Lagrangian \mathcal{L} is constant on a linear manifold of encoders $p(\hat{x}|x)$ [9] (Section 3.1), and so condition (9) leads to false detections. While this was considered there and in its sequel [19] by giving subtle conditions on the nullity of the second variation in (9), in practice it is difficult to tell whether a particular direction $\Delta q(\hat{x}|x)$ is in the kernel due to a bifurcation or due to other reasons, as they note. Second, note that a finite IB problem can be written as an *infinite* RD problem [20]. As discussed in Section 5, representing an IB root by a finite-dimensional vector leads to inherent subtleties in its computation. Among other things, these may well result in a bifurcation *not* being detectable under certain circumstances (Section 5.3). To our understanding, many of the difficulties that hindered the understanding of IB bifurcations throughout the years are, in fact, artifacts of finite dimensionality. Third, conditions (8) and (9) do not suffice to reveal the type of the bifurcation, information which is necessary for handling it when following a root's path. While [19] (Section 2.9) give conditions for identifying the type, these partially agree with our findings and do not suggest a straightforward way for handling a bifurcation.

Rather than imposing conditions on the scalar functional \mathcal{L} , our approach to IB bifurcations follows that of [6] for RD. That is, we rely on the fact that the IB's local extrema are fixed points of an iterative algorithm, and so they also satisfy a vector equation $F = \mathbf{0}$ (6). We shall now consider a toy problem to motivate our approach. “*Bifurcation Theory can be briefly described by the investigation of problem (6) in a neighborhood of a root where $D_x F$ is singular*” [21]. Indeed, recall that if $D_x F$ is non-singular at a root (x_0, β_0) , then by the Implicit Function Theorem (IFT), there exists a function $x(\beta)$ through the root, $x(\beta_0) = x_0$, which satisfies $F(x(\beta), \beta) = \mathbf{0}$ (6) at the vicinity of β_0 . The function $x(\beta)$ is then not only unique at some neighborhood of (x_0, β_0) , but further, $x(\beta)$ inherits the differentiability properties of F [21] (I.1.7). In particular, if the operator F is real-analytic in its variables—as with the IB operator (5)—then so is its root $x(\beta)$. While a bifurcation can occur only if $D_x F$ is singular, singularity is not sufficient for a bifurcation to occur. For example, the roots of the operator

$$F(x, y; \beta) := (x - \beta, 0) \quad (10)$$

on \mathbb{R}^2 consist of the vertical line $x = \beta$, $\{(\beta, y) : y \in \mathbb{R}\}$, for every $\beta \in \mathbb{R}$. For a fixed y , each such root is real-analytic in β . However, one cannot deduce this directly from the IFT, as the Jacobian $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ of F (10) is always singular. Note, however, that in this particular example, the x coordinate alone suffices to describe the problem's dynamics, and so its y coordinate is redundant. One can ignore the y coordinate by considering the “reduction” $\tilde{F}(x; \beta) := x - \beta$ of F to \mathbb{R}^1 . Further, discarding y also removes or *mods-out* the direction $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ from $\ker D_x F$, which does not pertain to a bifurcation in this case. This results in the non-singular Jacobian matrix (1) of \tilde{F} , and so it is now possible to invoke the IFT on the reduced problem. The root guaranteed by the IFT can always be considered in \mathbb{R}^2 by putting back a redundant y coordinate at some fixed value. In [6], a similarly defined *reduction* of finite RD problems was used to show that their dynamics are piecewise real-analytic under mild assumptions.

The intuition behind our approach is similar to [20] (Section III), who observed that “*in the IB one can also get rid of irrelevant variables within the model*”. Nevertheless, the details differ. Mathematically, we consider the *quotient* V/W of a vector space V by its subspace W . Elements of V are identified in the quotient if they differ by an element of W : $v_1 \sim v_2 \Leftrightarrow v_1 - v_2 \in W$, for $v_1, v_2 \in V$. This way, one “*mods-out*” W , collapsing it to a single point in the quotient vector space V/W . The resulting problem is smaller and so easier to handle, whether for theoretical or practical purposes (although not needed

for our purposes, this can be made precise in terms of the *tangent space* of a differentiable manifold; cf., Section 3 in [22]). This is how the one-dimensional vector space $\ker D_x F$ in our toy example (10) was reduced to the trivial $\ker D_x \tilde{F} = \{0\}$. However, one needs to understand the solution structure, for example, to ensure that the directions in W are not due to a bifurcation. We note in passing that V/W has a simple geometric interpretation as the translations of W in V , in a manner reminiscent of its better-known counterparts of quotient groups and rings; e.g., [23] (Section 10.2). To keep things simple, however, we shall not use quotients explicitly. Instead, the reader may simply consider the sequel as a removal of redundant coordinates, for we shall only remove coordinates that the reader does not care about anyway, as in the above toy example.

To achieve this approach, one needs to consider the IB in a coordinate system that permits a simple reduction as in (10), and to understand its solution structure. We achieve these in Section 5 by exploiting two properties of the IB which are often overlooked. First, proceeding with the coordinates' exchange of Section 2, the intimate relations [5,20] of the IB with RD suggest a "minimally sufficient" coordinate system for the IB, just as the x axis is for problem (10). Reducing an IB root to these coordinates is a natural extension of reduction in RD [6]. Reduction of IB roots facilitates a clean treatment of IB bifurcations. These are roughly divided into *continuous* and *discontinuous* bifurcations, in Sections 5.2 and 5.3, respectively. While understanding continuous bifurcations is straightforward, the IB's relations with RD allow us to understand the discontinuous bifurcation examples of which we are aware as a *support switching bifurcation* in RD, by leveraging [6] (Section 6). A second property is the analyticity of the IB operator (5), which stems from the analyticity of the IB Equations (2)–(4). By building on the first property, analyticity leads us to argue that the Jacobian of the IB operator (5) is generally non-singular (Conjecture 1) when considered in reduced coordinates as above. As an immediate consequence, the dynamics underlying the IB curve (1) are piecewise real-analytic in β , in a manner similar to RD. Indeed, the fact that there exist dynamics underlying the IB curve (1) in the first place can arguably be attributed to analyticity (see the discussion following Conjecture 1). Combining both properties sheds light on several subtle yet important practical caveats in solving the IB (Section 5.3) due to using finite-dimensional representations of its roots. These subtleties are compatible with our numerical experience. The results here suggest that, unlike RD, the IB is inherently infinite-dimensional, even for finite problems.

Finally, Section 6 combines the modified Euler method of Section 4 with the understanding of IB bifurcations in Section 5, to obtain an algorithm for following the path of an optimal IB root, in Section 6.1. That is, First-order Root Tracking for the IB (IBRT1). For simplicity, we focus mainly on continuous IB bifurcations, as these are the ones most often encountered in practice (see Section 6.3 on the algorithm's handling of discontinuous bifurcations). The resulting approximations in the information plane are surprisingly close to the true IB curve (1), even on relatively sparse grids (i.e., with large step sizes), as seen in Figure 1. See Section 6.2 for the numerical results underlying the latter. The reasons for this are discussed in Section 6.3, along with the algorithm's basic properties. Unlike BA-IB, which suffers from an increased computational cost near bifurcations, our IBRT1 algorithm suffers from a reduced accuracy there, in a manner similar to root tracking for RD [6].

With that, we note that there are standard techniques in Bifurcation Theory for handling a non-trivial kernel of $D_x F$ at a root. For example, the *Lyapunov–Schmidt reduction* replaces the high-dimensional problem $F = 0$ (6) on \mathbb{R}^n by a smaller but equivalent problem $\Phi = 0$, where $\Phi(\cdot, \beta)$ maps vectors in the (right) kernel of $D_x F$ to vectors in its left kernel. To achieve this, it separates the kernel and non-kernel directions of the problem, essentially handling each in turn; e.g., [21] (Theorem I.2.3) or [24] (Section 9.7). This technique is generic, as it does not rely on any particular property of the problem at hand. As such, it is considerably more involved than removing redundant coordinates, which requires an understanding of the solution structure. In contrast, reduction in the IB is straightforward. For the purpose of following a root's path, carrying on with redundant kernel directions is burdensome, computationally expensive, and sensitive to approximation errors. Applying

Lyapunov–Schmidt to our toy problem (10), for instance, reduces $F = \mathbf{0}$ (6) to choosing a continuously differentiable function Φ on the y -axis there (which is obtained by first solving for $x = \beta$; see the proof of Theorem I.2.3 in [21] for details). However, since y is redundant in this example, then solving for Φ can provide no useful information on the dynamics of its roots. In [19], a variant of the Lyapunov–Schmidt reduction was used to consider IB bifurcations due to symmetry breaking. While our findings are in partial agreement with theirs for continuous IB bifurcations, they differ for discontinuous bifurcations (see Sections 5.2 and 5.3).

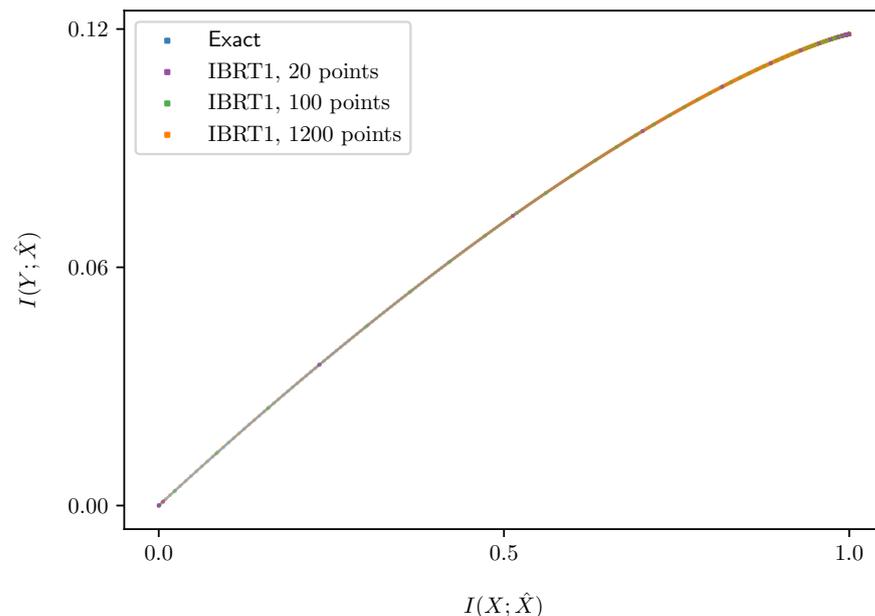


Figure 1. The approximate IB curves yielded by our algorithm, based on the IB ODE (16). Our First-order Root Tracking algorithm for the IB (IBRT1) of Section 6.1 was used to approximate the optimal IB roots of a binary symmetric channel with crossover probability 0.3 and a uniform source, BSC(0.3), for several grid densities. The points in the information plane yielded from these approximations are plotted on top of the problem’s exact solution (see Appendix E). Despite the algorithm’s approximation errors (Section 6.2), the approximate curves it yields are visually indistinguishable from the true IB curve (1), even on relatively few grid points. The reasons for this are discussed below (Section 6.3).

Notations

Vectors are written in boldface x , and scalars in regular font x . A distribution p pertaining to a particular multiplier value β of the IB Lagrangian \mathcal{L} is denoted with a subscript, p_β . Blahut–Arimoto’s algorithm for the IB (BA-IB) is brought below as Algorithm 1, with a single iteration over the IB Equations (2)–(4) (in steps 1.4–1.8) denoted BA_β . The *probability simplex* on a set S is denoted $\Delta[S]$ (see Section 5.1). The *support* of a probability distribution p on S is $\text{supp } p := \{s \in S : p(s) \neq 0\}$. The *source*, *label*, and *representation* alphabets of an IB problem are denoted \mathcal{X} , \mathcal{Y} , and $\hat{\mathcal{X}}$, respectively; we write $T := |\hat{\mathcal{X}}|$. δ denotes Dirac’s delta function, $\delta_{i,j} = 1$ if $i = j$, and zero otherwise.

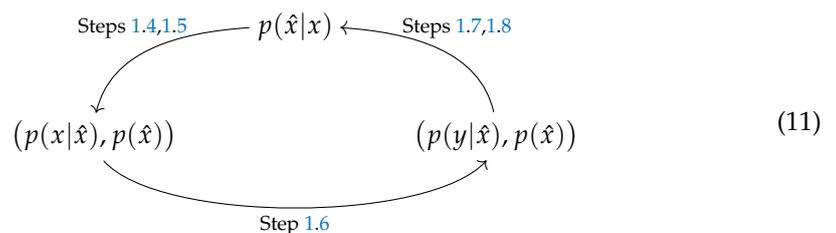
Algorithm 1 Blahut–Arimoto for the Information Bottleneck (BA-IB), [1].

1: **function** BA-IB($p_0(\hat{x}|x); p_{Y|X} p_X, \beta$)
Input:
 An initial encoder $p_0(\hat{x}|x)$, a problem definition $p(y|x)p(x)$, and $\beta > 0$.
Output:
 A fixed point $p(\hat{x}|x)$ of the IB Equations (2)–(4).
 2: Initialize $i \leftarrow 0$.
 3: **repeat**
 4: $p_i(\hat{x}) \leftarrow \sum_x p_i(\hat{x}|x)p(x)$
 5: $p_i(x|\hat{x}) \leftarrow p_i(\hat{x}|x)p(x)/p_i(\hat{x})$
 6: $p_i(y|\hat{x}) \leftarrow \sum_x p(y|x)p_i(x|\hat{x})$
 7: $Z_i(x, \beta) \leftarrow \sum_{\hat{x}} p_i(\hat{x}) \exp \{ -\beta D_{KL} [p(y|x)||p_i(y|\hat{x})] \}$
 8: $p_{i+1}(\hat{x}|x) \leftarrow \frac{p_i(\hat{x})}{Z_i(x, \beta)} \exp \{ -\beta D_{KL} [p(y|x)||p_i(y|\hat{x})] \}$
 9: $i \leftarrow i + 1$
 10: **until** convergence.
 11: **end function**

2. Coordinates Exchange for the IB

Just as a point in the plane can be described by different coordinate systems, so can IB roots. As demonstrated recently by [6] for the related rate-distortion theory, picking the right coordinates matters when analyzing its bifurcations. The same holds also for the IB. Our primary motivations for exchanging coordinates are to reduce computational costs and to mod-out irrelevant kernel directions, as explained in Section 1. In this Section, we discuss three natural choices of a coordinate system for parameterizing IB roots and the reasoning behind our choice for the sequel before setting to derive the IB ODE in the following Section 3. This work is complemented by the later Section 5.1, which facilitates a transparent analysis of IB bifurcations.

IB roots have been classically parameterized in the literature by (direct) encoders $p(\hat{x}|x)$, following [1]. Considering the BA-IB Algorithm 1 reveals two other natural choices, illustrated by Equation (11) below. First, an encoder $p(\hat{x}|x)$ determines a *cluster marginal* $p(\hat{x})$ and an *inverse encoder* $p(x|\hat{x})$, via steps 4 and 5 of Algorithm 1 (denoted 1.4 and 1.5, for short), respectively. These can be interpreted geometrically as $p(\hat{x})$ -weighted points $q_{\hat{x}}(x)$ in the simplex $\Delta[\mathcal{X}]$ of X , so long as they are well-defined, $\forall \hat{x} p(\hat{x}) \neq 0$. No more than $|\mathcal{X}| + 1$ points in the simplex are required to represent an IB root [2]. The latter is readily seen to analyze the IB in these coordinates, although it pre-dates [1] and has generally escaped broader attention. Second, an inverse encoder determines a *decoder* $p(y|\hat{x})$, via step 6. Along with the cluster marginal, $(p(y|\hat{x}), p(\hat{x}))$ can be similarly interpreted as $p(\hat{x})$ -weighted points $r_{\hat{x}}(y)$ in the simplex $\Delta[\mathcal{Y}]$ of Y . This choice of coordinates is implied already by Theorem 5 in [1]. Cycling around Equation (11), a decoder $(p(y|\hat{x}), p(\hat{x}))$ determines via steps 7 and 8 a new encoder, which may differ from the one with which we have started. For notational simplicity, we shall usually write $(p(y|\hat{x}), p(\hat{x}))$ rather than $(r_{\hat{x}}(y), p(\hat{x}))$ for decoder coordinates (similarly for inverse encoder coordinates).



The above allows us to define *three* BA operators as the composition of three consecutive maps in Equation (11), encoding an iteration of Algorithm 1. When starting at an encoder $p(\hat{x}|x)$, its output is a newly defined encoder. Similarly, when starting at one of the other two vertices, it sends an inverse encoder $(p(x|\hat{x}), p(\hat{x}))$ or a decoder pair $(p(y|\hat{x}), p(\hat{x}))$ to

a newly defined one. By abuse of notation, we denote all three compositions by BA_β , with the choice of coordinate system mentioned accordingly. Indeed, these are representations of a single BA-IB iteration in three different coordinate systems, and so may be considered as distinct representations of the same operator. For completeness, BA_β in decoder coordinates is spelled out explicitly in Equation (A1) in Appendix A. A newly defined encoder (or inverse encoder or decoder) at a cycle’s completion need not generally equal the one at which we have started. These are equal precisely at IB roots, when the IB Equations (2)–(4) hold. Therefore, the choice of a coordinate system does *not* matter then, and so moving around Equation (11) from one vertex to another yields different parameterizations of the same root, at least when $\forall \hat{x} p(\hat{x}) \neq 0$. In particular, this shows that the inverse encoders $q_{\hat{x}}$ in $\Delta[\mathcal{X}]$ of an IB root are in bijective correspondence with its decoders $r_{\hat{x}}$ in $\Delta[\mathcal{Y}]$, an observation which shall come in handy in Section 5.

Next, we consider how well each of these coordinate systems can serve for following the path of an IB root. The minimal number of symbols \hat{x} needed to write down an IB root typically varies with the constraints, cf., [1] (Section 3.4) or [2] (Section II.A). Therefore, inverse encoder and decoder coordinates are better suited than encoder coordinates for considering the dynamics of a root with β , as they allow us to consider its evolution via a varying number of points in a fixed space, $\Delta[\mathcal{X}]$ or $\Delta[\mathcal{Y}]$, respectively. Indeed, a direct encoder $p(\hat{x}|x)$ can be interpreted geometrically as a point in the $|\mathcal{X}|$ -fold product $\Delta[\hat{\mathcal{X}}]^\mathcal{X}$ of simplices $\Delta[\hat{\mathcal{X}}]$ [9] (Section 2). So, if a particular symbol \hat{x}' is not in use anymore, $p(\hat{x}') = 0$, then one is forced to choose between replacing $\Delta[\hat{\mathcal{X}}]$ by a smaller space $\Delta[\hat{\mathcal{X}} \setminus \{\hat{x}'\}]$ or carrying on with a redundant symbol \hat{x}' . The latter leads to non-trivial kernels in the IB due to duplicate clusters (e.g., Section 3.1 there), making it difficult to tell whether a particular kernel direction pertains to a bifurcation (or to a “perpetual kernel” [9,19]). In contrast, when considered in decoder coordinates, for example, an IB root is nothing but $p(\hat{x})$ -weighted paths r_1, \dots, r_T in $\Delta[\mathcal{Y}]$, with $\beta \mapsto r_{\hat{x}}(\beta)$ a path for each \hat{x} . And so, once a symbol \hat{x}' is not needed anymore, then one can discard the path $r_{\hat{x}'}$ without replacing the underlying space $\Delta[\mathcal{Y}]$. This permits the clean treatment of IB bifurcations in Section 5.

The computational cost of solving a first-order ODE as in (7) numerically in $\frac{dx}{d\beta}$ depends on $\dim x$. Much of this cost is due to computing a linear pre-image under $D_x F$, which is of order $O(\dim x)^3$ [25] (Section 28.4); cf., Section 6. Representing an IB root on T clusters in encoder coordinates requires $|\mathcal{X}| \cdot T$ dimensions (ignoring normalization constraints), in inverse encoder coordinates $(|\mathcal{X}| + 1) \cdot T$ dimensions, and in decoder coordinates $(|\mathcal{Y}| + 1) \cdot T$ dimensions. Thus, the computational cost is lowest in decoder coordinates, at least when there are fewer possible labels \mathcal{Y} than input symbols \mathcal{X} .

A priori, one might expect that derivatives with respect to β vanish when the solution barely changes, regardless of the choice of coordinate system. For example, at a very large “ $\beta = \infty$ ” value, an obvious IB root is the diagonal encoder (setting $\hat{\mathcal{X}} := \mathcal{X}$ and $p(\hat{x}|x) := \delta_{x,\hat{x}}$), as can be seen by a direct examination of the IB Equations (2)–(4). It consists of one IB cluster of weight (or *mass*) $p(x)$ at $p_{Y|X=x} \in \Delta[\mathcal{Y}]$ for each $x \in \mathcal{X}$, and so one might expect that it would barely change so long as β is very large. However, the logarithmic derivative $\frac{d \log p_\beta(\hat{x}|x)}{d\beta}$ in encoder coordinates need *not* vanish even when the derivatives $\frac{d \log p_\beta(y|\hat{x})}{d\beta}$ and $\frac{d \log p_\beta(\hat{x})}{d\beta}$ in decoder coordinates do (see Section 3 on logarithmic coordinates), as seen to the right of Figure 2. Indeed, given the derivative in decoder coordinates, one can exchange it to encoder coordinates by

$$\begin{aligned} \frac{d \log p_\beta(\hat{x}|x)}{d\beta} &= J_{\text{dec}}^{\text{enc}} \frac{d \log p_\beta(y'|\hat{x}')}{d\beta} + J_{\text{mrg}}^{\text{enc}} \frac{d \log p_\beta(\hat{x}')}{d\beta} \\ &\quad - D_{KL}[p(y|x)||p_\beta(y|\hat{x})] + \sum_{\hat{x}''} p_\beta(\hat{x}''|x) D_{KL}[p(y|x)||p_\beta(y|\hat{x}'')] , \end{aligned} \quad (12)$$

where $J_{\text{dec}}^{\text{enc}}$ and $J_{\text{mrg}}^{\text{enc}}$ are the two coordinate exchange Jacobian matrices of orders $(T \cdot |\mathcal{X}|) \times (T \cdot |\mathcal{Y}|)$ and $(T \cdot |\mathcal{X}|) \times T$, respectively, given by Equations (A68) and (A70) in

Appendix B.4.2. And so, $\frac{d \log p_\beta(\hat{x}|x)}{d\beta}$ would often be non-zero even if both $\frac{d \log p_\beta(y|\hat{x})}{d\beta}$ and $\frac{d \log p_\beta(\hat{x})}{d\beta}$ vanish. This unintuitive behavior of the derivative in encoder coordinates is due to the explicit dependence of the IB’s encoder Equation (2) on β . This dependence is the source of the last two terms in Equation (12) (see Equation (A73)). The comparison between encoder and inverse encoder coordinates can be seen to be similar. See Appendix B.4 for further details.

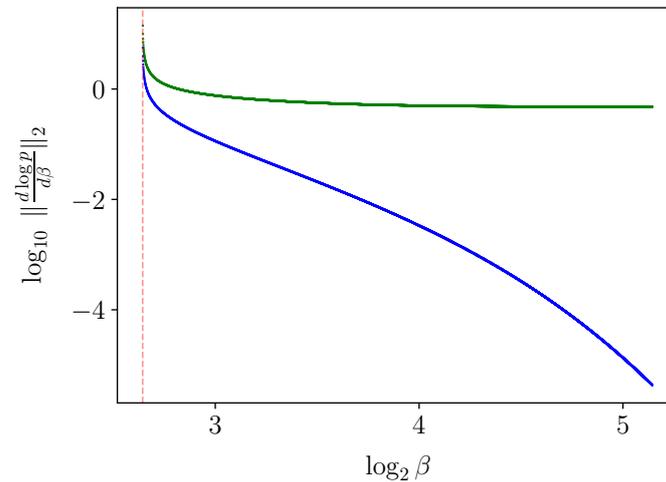


Figure 2. Derivatives’ norm by coordinate system, for the exact solution of BSC(0.3) with a uniform source, as in Figure 1; see Appendix E. The derivative’s L_2 -norm is plotted in green for encoder coordinates and blue for decoder coordinates. The solution barely changes at high β values, and so the derivative in *decoder* coordinates is smaller (see main text). Nevertheless, the derivative in *encoder* coordinates does not vanish then, due to Equation (12). At low β values, however, the derivative in either coordinate system may generally be large. Both vanish to the left of the bifurcation in this problem (dashed red vertical), as the solution there is trivial (single-clustered). The derivatives diverge near the bifurcation (to its right) regardless of the coordinate system, as might be expected by the implicit ODE (7)—see also Section 6.1.

In light of the above, we proceed with decoder coordinates in the sequel.

3. Implicit Derivatives at an IB Root and the IB’s ODE

We now specialize the implicit ODE (7) (of Section 1) to the IB, using the decoder coordinates of the previous Section 2. This allows us to compute first-order implicit derivatives at an IB root (Theorem 1) with remarkable accuracy, under one primary assumption—that the root is a differentiable function of β . While differentiability breaks at IB bifurcations (Section 5), this allows us to reconstruct a solution path from its local approximations in the following Section 4, so long as it holds.

To simplify calculations, we take the logarithm $(\log p(y|\hat{x}), \log p(\hat{x}))$ of the decoder coordinates of Section 2 as our variables. Exchanging the BA_β operator to log-decoder coordinates is immediate, by writing $\log BA_\beta[\exp(\log p(y|\hat{x})), \exp(\log p(\hat{x}))]$. For short, we denote it $BA_\beta[\log p(y|\hat{x}), \log p(\hat{x})]$ when in these coordinates, by abuse of notation. Similarly, exchanging the IB ODE (below) back to non-logarithmic coordinates is immediate, via $\frac{d}{d\beta} \log p = \frac{1}{p} \frac{d}{d\beta} p$. In Section 6, we shall assume that $p(\hat{x})$ never vanishes. To ensure that taking logarithms is well-defined, we require that no decoder coordinate $p(y|\hat{x})$ vanishes (while it may have a well-defined derivative $\frac{d}{d\beta} p(y|\hat{x})$ even with a vanished coordinate, calculation details would differ). A sufficient condition for that is that $p(y|x) > 0$ for every x and y (Lemma A1 in Appendix A).

Next, define a variable $x \in \mathbb{R}^{T \cdot (|\mathcal{Y}|+1)}$ as the concatenation of the vector $(\log p_\beta(y|\hat{x}))_{y \in \mathcal{Y}, \hat{x} \in \hat{\mathcal{X}}}$ with $(\log p_\beta(\hat{x}))_{\hat{x} \in \hat{\mathcal{X}}}$. Differentiating $\partial/\partial \log p$ with respect to log-

probabilities is given by $p \cdot \frac{\partial}{\partial p}$, by the chain rule (setting $u := \log p$, $\frac{df(p)}{du} = \frac{df}{dp} \frac{dp}{du}$, or equivalently $\frac{df}{d \log p} = p \cdot \frac{df}{dp}$; see Appendix B.1 for a gentler treatment). This gives meaning to the Jacobian matrix $D_x(\cdot)$ with respect to our logarithmic variable x . The Jacobian $D_{\log p(y|\hat{x}), \log p(\hat{x})} BA_\beta$ of a single Blahut–Arimoto iteration in these log-decoder coordinates is a square matrix of order $T \cdot (|\mathcal{Y}| + 1)$. Its $(T \cdot |\mathcal{Y}|) \times (T \cdot |\mathcal{Y}|)$ upper-left block (below) corresponds to perturbations of BA’s output log-decoder $\log p(y|\hat{x})$ due to varying an input log-decoder $\log p(y'|\hat{x}')$. Since we prime input but not output coordinates, this is to say that the *columns* of this block are indexed by pairs (y', \hat{x}') and its *rows* by (y, \hat{x}) (one could also enumerate $\mathcal{Y} := \{y_1, \dots, y_{|\mathcal{Y}|}\}$ and $\hat{\mathcal{X}} := \{\hat{x}_1, \dots, \hat{x}_T\}$ explicitly, replacing (y, \hat{x}) and (y', \hat{x}') throughout by (y_i, \hat{x}_j) and (y_k, \hat{x}_l) , respectively, with $i, k = 1, \dots, |\mathcal{Y}|$ and $j, l = 1, \dots, T$). Its $(T \cdot |\mathcal{Y}|) \times T$ upper-right block corresponds to perturbations in BA’s output log-decoder $\log p(y|\hat{x})$ due to varying an input log-marginal $\log p(\hat{x}')$. That is, its columns are indexed by \hat{x}' and rows by (y, \hat{x}) . Similarly, for the bottom-left and bottom-right blocks, of respective sizes $T \times (T \cdot |\mathcal{Y}|)$ and $T \times T$. See (A25) ff., in Appendix B.2, and the end-result at Equation (A44) there. Explicitly, when evaluated at an IB root $(\log p(y|\hat{x}), \log p(\hat{x}))$, BA’s Jacobian matrix is given by

$$D_{\log p(y|\hat{x}), \log p(\hat{x})} BA_\beta [\log p(y|\hat{x}), \log p(\hat{x})] = \left(\begin{array}{c|c} \beta \cdot \sum_{\hat{x}'', y''} (\delta_{\hat{x}'', \hat{x}'} - \delta_{\hat{x}, \hat{x}'}) \cdot \left[1 - \frac{\delta_{y'', y}}{p_\beta(y|\hat{x})} \right] C(\hat{x}, \hat{x}''; \beta)_{y', y''} & (1 - \beta) \cdot \sum_{y''} \left[1 - \frac{\delta_{y'', y}}{p_\beta(y|\hat{x})} \right] B(\hat{x}, \hat{x}'; \beta)_{y''} \\ \hline \beta \cdot \left[\delta_{\hat{x}, \hat{x}'} p_\beta(y'|\hat{x}) - B(\hat{x}, \hat{x}'; \beta)_{y'} \right] & (1 - \beta) \cdot \left[\delta_{\hat{x}, \hat{x}'} - A(\hat{x}, \hat{x}'; \beta) \right] \end{array} \right) \tag{13}$$

where $\delta_{i,j} = 1$ if $i = j$ and is 0 otherwise. As mentioned above, primed coordinates y' and \hat{x}' index the columns, and un-primed coordinates y and \hat{x} the rows. Indices y'' and \hat{x}'' with more than a single prime are summation variables. A , B , and C are a scalar, a vector, and a matrix, each involving two IB clusters. They are defined by

$$\begin{aligned} A(\hat{x}, \hat{x}'; \beta) &:= \sum_{x''} p_\beta(\hat{x}'|x'') p_\beta(x''|\hat{x}), \\ B(\hat{x}, \hat{x}'; \beta)_y &:= \sum_{x''} p(y|x'') p_\beta(\hat{x}'|x'') p_\beta(x''|\hat{x}), \quad \text{and} \\ C(\hat{x}, \hat{x}'; \beta)_{y, y'} &:= \sum_{x''} p(y|x'') p(y'|x'') p_\beta(\hat{x}'|x'') p_\beta(x''|\hat{x}). \end{aligned} \tag{14}$$

In these, y indexes B and the rows of C , y' the columns of C , and x'' is a summation variable. These (\hat{x}, \hat{x}') -labeled tensors have only $|\mathcal{Y}|$ entries along each axis, thanks to our choice of decoder coordinates. A and B can be expressed in terms of C via some obvious relations; see Equation (A32) and below in Appendix B.2. Appendix B.1 elaborates on the mathematical subtleties involved in calculating the Jacobian (13). See also Equation (A45) in Appendix B.2 for an implementation-friendly form of (13).

Together with $D_\beta BA_\beta$ (Equations (A58) and (A57) in Appendix B.3), we have both of the first-order derivative tensors of BA_β in log-decoder coordinates. This allows us to specialize the implicit ODE (7) (of Section 1) to the IB, in terms of our variable x . By abuse of notation, we write $(\log p_\beta(y|\hat{x}), \log p_\beta(\hat{x}))_{y, \hat{x}}$ for its $|\mathcal{Y}| \cdot T + T$ coordinates, and similarly for its derivatives vector v (15) below.

Theorem 1 (The IB’s ODE). *Let $(p(y|\hat{x}), p(\hat{x}))$ be an IB root, and suppose that it can be written as a differentiable function $\beta \mapsto (p_\beta(y|\hat{x}), p_\beta(\hat{x}))$ in β . If none of its coordinates vanish, then the vector*

$$v := \left(\frac{d \log p_\beta(y|\hat{x})}{d\beta}, \frac{d \log p_\beta(\hat{x})}{d\beta} \right)_{y, \hat{x}} \tag{15}$$

of its implicit logarithmic derivatives is well-defined and satisfies an ordinary differential equation in β ,

$$\left(I - D_{\log p(y|\hat{x}), \log p(\hat{x})} B A_\beta \right) \mathbf{v} = \begin{pmatrix} -\sum_{x, x''} \left[1 - \frac{p(y|x)}{p_\beta(y|\hat{x})} \right] \cdot \left[\delta_{\hat{x}, x''} - p_\beta(x''|x) \right] p_\beta(x|\hat{x}) D_{KL} [p(y|x) || p_\beta(y|\hat{x}'')] \\ \sum_{x, x''} \left[\delta_{\hat{x}, x''} - p_\beta(x''|x) \right] p_\beta(x|\hat{x}) D_{KL} [p(y|x) || p_\beta(y|\hat{x}'')] \end{pmatrix} \quad (16)$$

where I is the identity matrix of order $T \cdot (|\mathcal{Y}| + 1)$, and the Jacobian matrix $D_{\log p(y|\hat{x}), \log p(\hat{x})} B A_\beta$ at the given IB root is given by Equation (13). The right-hand side of (16) is indexed as in (15), by (y, \hat{x}) at its top and \hat{x} at its bottom coordinates.

While the IB ODE was discovered by [10], it is derived here anew in log-decoder coordinates due to the considerations in Section 2. It is analogous to the RD ODE, due to [6]; Corollary 1 and around (in Section 5.1) provides a relation between these two ODEs. We emphasize that the first assumption of Theorem 1, that the IB root is a differentiable function of β , is essential. It consists of two parts: (i) that the root can be written as a function of β , and (ii) that this function is differentiable. These are precisely the assumptions needed to compute the first-order implicit multivariate derivative \mathbf{v} (15) at the given root [6] (Section 2.1). Continuous IB bifurcations violate (ii) (Section 5.2), while discontinuous ones violate (i) (Section 5.3). In contrast, the requirement that no coordinate vanishes is a technical one, due to our choice of logarithmic coordinates.

It is not necessary for the Jacobian of the IB operator (5) (to the left of (16)) to be non-singular in order to solve the IB ODE numerically. Nevertheless, non-singularity of the Jacobian will follow from the sequel (see Conjecture 1 in Section 5). With that, the derivatives $\mathbf{v} = \frac{d}{d\beta} (\log p_\beta(y|\hat{x}), \log p_\beta(\hat{x}))$ (15) computed numerically from the IB ODE (16) at an exact root are remarkably accurate, as demonstrated in Figure 3. As in RD [6], calculating implicit derivatives numerically loses its accuracy when approaching a bifurcation because the Jacobian is increasingly ill-conditioned there. For comparison, the BA-IB Algorithm 1 also loses its accuracy near a bifurcation. This is a consequence of BA's critical slowing down [4], just as with its corresponding RD variant.

Each coordinate of $(p(y|\hat{x}), p(\hat{x}))$ is treated by the IB ODE (16) as an independent variable. However, the normalization of $p(y|\hat{x})$ imposes one constraint per cluster \hat{x} (and one for the normalization of $p(\hat{x})$). Thus, one might expect the behavior of BA's Jacobian (13) to be determined by fewer than $T \cdot (|\mathcal{Y}| + 1)$ coordinates, at least qualitatively. This intuition is justified by the following Lemma 1, which allows us to consider the kernel of the IB operator (5) by a smaller and simpler matrix S ; see Appendix C for its proof.

Lemma 1. Given an IB root as above, define a square matrix of order $T \cdot |\mathcal{Y}|$ by

$$S_{(y, \hat{x}), (y', \hat{x}')} := \sum_x p_\beta(x|\hat{x}) \left[\beta \cdot \frac{p(y|x)}{p_\beta(y|\hat{x})} + (1 - 2\beta) \right] p(y'|x) \left[\delta_{\hat{x}, \hat{x}'} - p_\beta(\hat{x}'|x) \right]. \quad (17)$$

Then, the nullity of the Jacobian $I - D_{\log p(y|\hat{x}), \log p(\hat{x})} B A_\beta$ of the IB operator (5) equals that of $I - S$, where I is the identity matrix (of the respective order), and S is defined by (17),

$$\dim \ker(I - S) = \dim \ker \left(I - D_{\log p(y|\hat{x}), \log p(\hat{x})} B A_\beta \right). \quad (18)$$

Specifically, write $\mathbf{v} := (v_{y, \hat{x}})_{y, \hat{x}}$ for a left eigenvector which corresponds to $1 \in \text{eig } S$. Then, there is a bijective correspondence between the left kernels at both sides of (18), mapping

$$\mathbf{v} \mapsto (\mathbf{v}, \mathbf{u}), \quad (19)$$

where $\mathbf{u} := (u_{\hat{x}})_{\hat{x}}$ is defined by $u_{\hat{x}} := \frac{1-\beta}{\beta} \cdot \sum_y v_{y, \hat{x}}$.

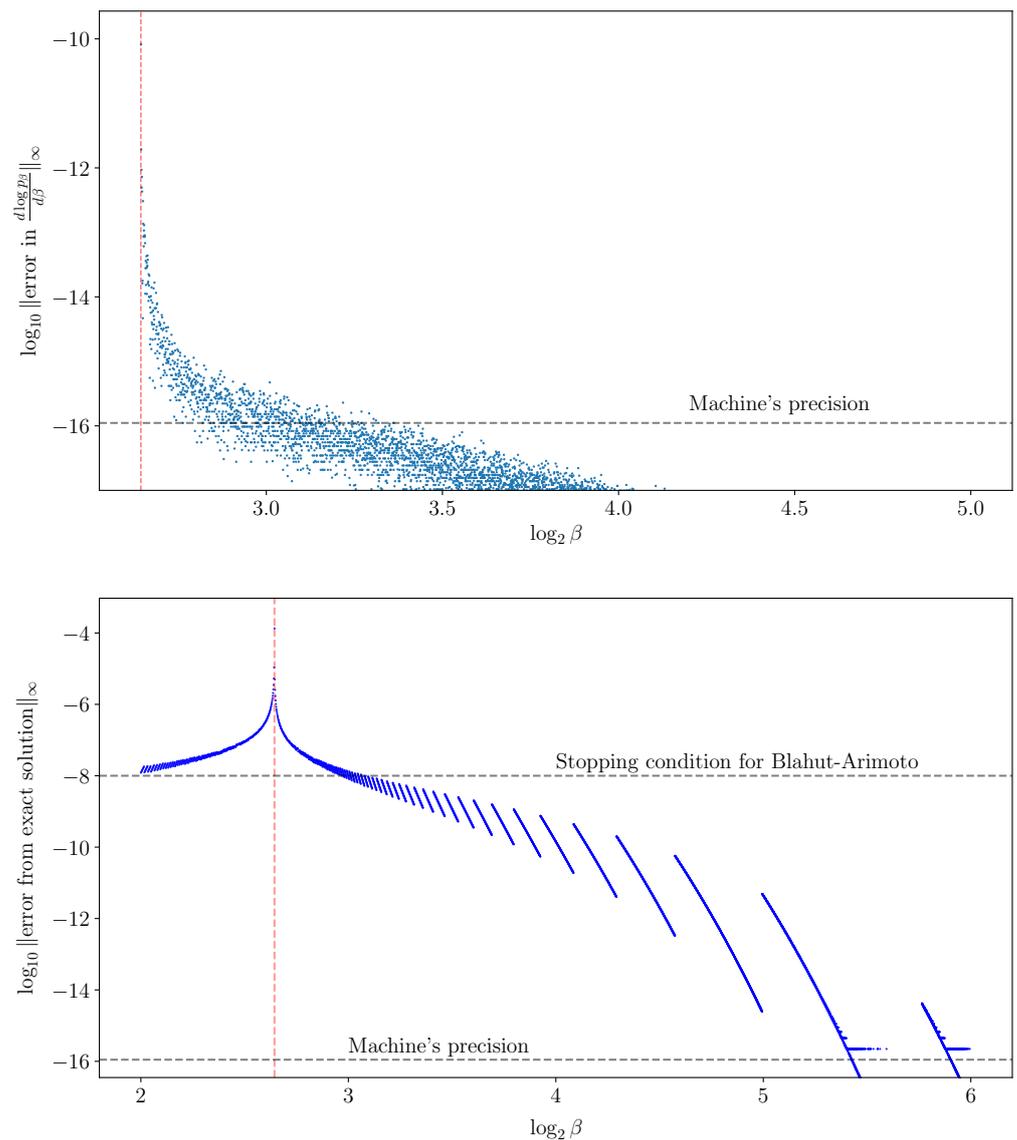


Figure 3. The implicit derivatives computed from the IB ODE (16) are very accurate, as is the BA-IB Algorithm 1. However, both lose their accuracy near a bifurcation. To verify their accuracy, we compared both to the exact solutions of BSC(0.3) with a uniform source (see Appendix E). (Top): Derivatives were computed at the problem’s exact solution using the IB ODE (16) and compared to the problem’s exact derivatives. These are accurate beyond the machine’s precision, except when approaching the bifurcation (red vertical), since the Jacobian of the IB operator (5) is ill-conditioned there. (Bottom): The L_∞ -errors of the solutions produced by the BA-IB Algorithm 1, with a 10^{-8} stopping condition, and uniform initial conditions. Error is measured from the true direct encoder to avoid biases due to clusters of low mass. Both plots are as in Figure 2.3 of [6].

In addition to offering a form more transparent than BA’s Jacobian in (13), Lemma 1 also reduces the computational cost of testing $I - D_{\log p(y|\hat{x}), \log p(\hat{x})} BA_\beta$ (16) for singularity, by using the smaller $I - S$ (17) in its place. This makes it easier to detect upcoming bifurcations (see Conjecture 1 in Section 5). Further, one can verify directly that the IB ODE (16) indeed follows the right path. Indeed, if the ODE is non-singular, then, by the Implicit Function Theorem, there is (locally) a unique IB root, which is a differentiable function of β . And so, there is a unique solution path for a numerical approximation to follow. Finally, we note that a relation similar to (18) holds also for eigenvalues of $D_{\log p(y|\hat{x}), \log p(\hat{x})} BA_\beta$ (13) other than 1. This can be seen either empirically or by tracing the proof of Lemma 1.

In Section 5, we shall proceed with this line of thought of removing redundant coordinates. In the following Section 4, we turn to reconstruct a solution path from implicit derivatives at a point, with bifurcations ignored for now.

4. A Modified Euler Method for the IB

We follow the path of a given IB root away from bifurcation by using its implicit derivatives computed from the IB ODE (16), of Section 3. We follow the classic Euler method for simplicity, modifying it slightly to get the most out of the calculated derivatives. Improvements using more sophisticated numerical methods are left to future work. The detection and handling of IB bifurcations are deferred to the following Section 5, and thus are ignored in this section.

Let $\frac{dx}{d\beta} = f(x, \beta)$ and $x(\beta_0) = x_0$ define an initial value problem. In numerical approximations of ordinary differential equations (ODEs), the *Euler method* for this problem is defined by setting

$$x_{n+1} := x_n + \Delta\beta \cdot f(x_n, \beta_n), \tag{20}$$

where $\beta_{n+1} := \beta_n + \Delta\beta$, and $|\Delta\beta|$ is the *step size*. The *global truncation error* $\max_n \|x_n - x(\beta_n)\|_\infty$ is the largest error of the approximations x_n from the true solutions $x(\beta_n)$. A numerical method for solving ODEs is said to be of *order* d if its global truncation error is of order $O(|\Delta\beta|^d)$, for step sizes $|\Delta\beta|$ small enough. Euler’s method error analysis is a standard result, provided as Theorem 2 below. See [26] (Theorem 212A) or [27] (Theorem 2.4), for example. It shows that Euler’s method (20) is of order $d = 1$, under mild assumptions, as demonstrated in Figure 4. The immediate generalization of (20) using derivatives until order d is *Taylor’s method*, which is a method of order d .

Theorem 2 (Euler’s method error analysis). *Let an initial value problem be defined on $[\beta_0, \beta_f]$ by $\frac{dx}{d\beta} = f(x, \beta)$ as above (with x_0 allowed to deviate from $x(\beta_0)$), and suppose that f satisfies the Lipschitz condition with some constant $L > 0$. Namely, $\|f(x, \beta) - f(x', \beta)\|_\infty \leq L \cdot \|x - x'\|_\infty$ for every x, x' and $\beta \in [\beta_0, \beta_f]$.*

Then, Euler’s method (20) global truncation error satisfies

$$\max_{\beta_0 \leq \beta_n \leq \beta_f} \|x_n - x(\beta_n)\|_\infty \leq e^{(\beta_f - \beta_0)L} \|x_0 - x(\beta_0)\|_\infty + \frac{e^{(\beta_f - \beta_0)L} - 1}{L} \cdot \frac{1}{2} |\Delta\beta| \max_{\beta_0 \leq \beta \leq \beta_f} \left\| \frac{d^2 x(\beta)}{d\beta^2} \right\|_\infty. \tag{21}$$

Specializing Euler’s method to our needs, replace x in (20) above by the log-decoder coordinates of an IB root, as in Section 3. So long as an IB root $p_\beta := (p_\beta(y|\hat{x}), p_\beta(\hat{x}))$ is a differentiable function of β in the vicinity of β_n , it can be approximated by

$$\begin{aligned} \log p_{\beta_{n+1}}(y|\hat{x}) &\approx \log p_{\beta_n}(y|\hat{x}) + \Delta\beta \cdot \left. \frac{d \log p_\beta(y|\hat{x})}{d\beta} \right|_{p_{\beta_n}} \quad \text{and} \\ \log p_{\beta_{n+1}}(\hat{x}) &\approx \log p_{\beta_n}(\hat{x}) + \Delta\beta \cdot \left. \frac{d \log p_\beta(\hat{x})}{d\beta} \right|_{p_{\beta_n}}, \end{aligned} \tag{22}$$

where $\frac{d \log p_\beta(y|\hat{x})}{d\beta}$ and $\frac{d \log p_\beta(\hat{x})}{d\beta}$ are calculated from the IB ODE (16). Thus, applying (22) repeatedly, we obtain an Euler method for the IB. We shall take only negative steps $\Delta\beta < 0$ when approximating the IB, due to reasons explained in Section 5.3 (after Proposition 1). In contrast to the BA-IB Algorithm 1, Euler’s method (22) can be used to interpolate intermediate points, yielding a piecewise linear approximation of the root.

The problem of tracking an operator’s root belongs in general to a family of hard-to-solve numerical problems—known as *stiff*—if the problem has a bifurcation [6] (Section 7.2). See [26] or [27] for example on stiff differential equations. Stopping early in the vicinity of a bifurcation restricts the computational difficulty and permits convergence guarantees. Early stopping in the IB shall be handled later, in Section 5.2. [6] (Theorem 5) proves that

Euler’s method convergence guarantees (Theorem 2) hold for the closely related Euler method for RD with early stopping. While Euler’s method may inadvertently switch between solution branches of the IB ODE (16), the latter guarantees ensure that it indeed follows the true solution path between bifurcations, if the step size $|\Delta\beta|$ is small enough and initializing close enough to the true solution (see Sections 5 and 6.3 on the distinction between IB bifurcations and singularities of the IB ODE (16)). Although we do not dive into these details for brevity, we note that similar convergence guarantees can also be proven here. Alternatively, Euler’s method can be ensured to follow the true solution path by noting that an optimal IB root is (strongly) stable when negative steps $\Delta\beta < 0$ are taken; these details are deferred to Section 6.3, as they depend on Section 5.

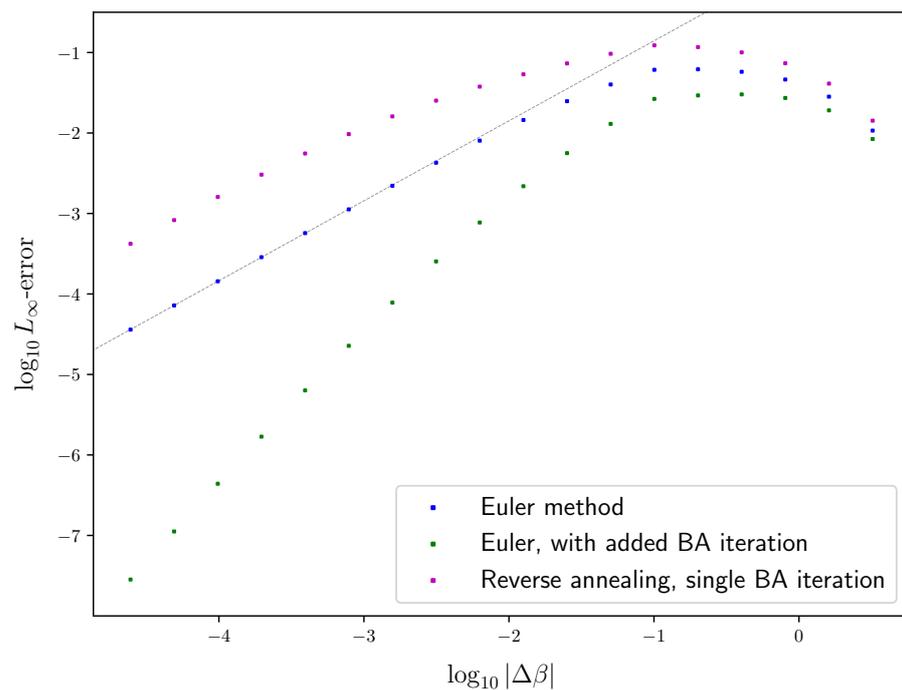


Figure 4. Error by step-size for a vanilla Euler method using the IB ODE (16), and with an added BA-IB iteration at each step, for BSC(0.3) with a uniform source (Appendix E). The linear regression (dashed black) of the third leftmost markers for the vanilla Euler method is of slope 0.99 ($R^2 \simeq 1$), matching the theory’s prediction almost perfectly. A similar regression (not shown) for Euler’s method with a single added BA iteration is of nearly double slope 1.93. For comparison, reverse deterministic annealing with a single BA iteration at each grid point yields a slope of 0.91 in this example. Taking a larger (pre-determined) number of iterations at each grid point pushes the error downwards, as expected. Yet, the resulting slopes approach 1 as the number of iterations is increased (not shown). See main text and Appendix D for details. The error was calculated as the supremum of the pointwise errors as in Figure 3, over the interval $[\beta_c + \frac{1}{10}, \beta_0]$ which contains no bifurcation. Each method was initialized with the exact solution at $\beta_0 = 2^5$, with $\Delta\beta = -\frac{103}{32}$ halved between consecutive markers.

Following the discussion in Section 2, there is a subtle disadvantage in choosing decoder coordinates as our variables compared to the other two coordinate systems there. Indeed, recall that the IB is defined as a maximization over Markov chains $Y \longleftrightarrow X \longleftrightarrow \hat{X}$. An (arbitrary) encoder $p(\hat{x}|x)$ defines a joint probability distribution $p(\hat{x}|x)p(y|x)p(x)$ which is Markov. An inverse encoder pair also similarly defines a Markov chain. In contrast, an arbitrary decoder pair $(p(y|\hat{x}), p(\hat{x}))$ need *not* necessarily define a Markov chain. Rather, by invoking the error analysis of Euler’s method, one can see that Markovity is approximated at an increasingly improved quality as the step-size $|\Delta\beta|$ in (22) becomes smaller. To enforce Markovity, we shall perform a single BA iteration (in decoder coordi-

nates) after each Euler method step. This ensures that the newly generated decoder pair satisfies the Markov condition, as it is now generated from an encoder.

As a side effect, adding a single BA-IB iteration after each Euler method step improves the approximation's quality significantly. By linearizing BA_β around a fixed point, one can show that deterministic annealing with a fixed number of BA iterations per grid point is a first-order method. Thus, deterministic annealing may arguably be considered a first-order method, as is with Euler's method. A similar argument shows that adding a single BA iteration after each Euler method step yields a second-order method. However, while a larger number of added BA iterations obviously improves the approximation's quality, it does not improve the method's order. See Appendix D for an approximate error analysis. The predicted orders are in good agreement with the ones found empirically, shown in Figure 4. We note that while [6] did not attempt an added BA iteration, they do discuss a variety of other improvements to root tracking (see Section 3.4 in [6]).

5. On IB Bifurcations

For the IB Equations (2)–(4) to exhibit a bifurcation, it is necessary that the Jacobian of the IB operator (5) be singular, as illustrated by Figure 5. However, a priori singularity is not sufficient to detect a bifurcation (cf., Section 3.1 in [9]), nor does this allow one to distinguish between bifurcations of different types. At an IB root, singularities of the IB ODE (16) (Section 3) coincide with those of $Id - BA_\beta$ (5) (in log-decoder coordinates). Thus, in order to be able to exploit the IB ODE (16), we shall now take a closer look into IB bifurcations. These can be broadly classified into two types: where an optimal root is continuous in β and where it is not. As noted after Theorem 1, each type violates an assumption necessary to compute implicit derivatives. Sections 5.2 and 5.3 provide the means to identify bifurcations, distinguish between their types, and handle them accordingly, mainly for continuous bifurcations. To facilitate the discussion, Section 5.1 considers the IB as a rate-distortion problem, following [20] and others. This allows us to leverage recent insights on RD bifurcations [6], while suggesting a “minimally sufficient” choice of coordinates for the IB. The latter permits a clean treatment of continuous IB bifurcations in Section 5.2. Viewing the IB as an infinite-dimensional RD problem facilitates the understanding of its discontinuous bifurcations, which in turn highlight subtleties in its finite-dimensional coordinate systems (of Section 2). These provide insight into the IB and are also of practical implications (Section 5.3), and so are necessary for our algorithms in Section 6.

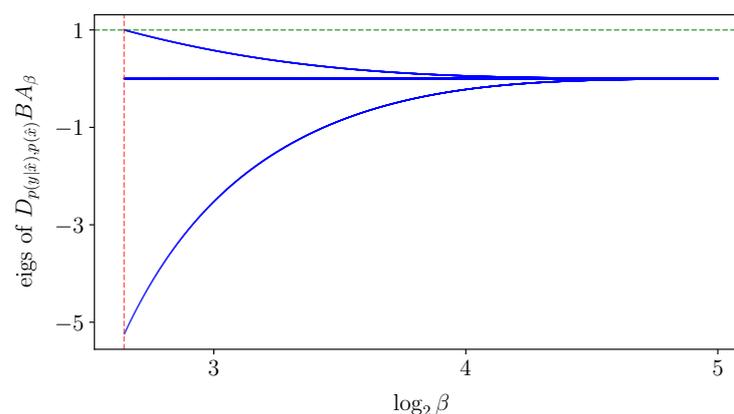


Figure 5. While the Jacobian $D_{\log p(y|\hat{x}), \log p(\hat{x})}(Id - BA_\beta)$ must be singular at a bifurcation, this does not suffice to identify its type. The Jacobian eigenvalues of BA_β (13) with respect to log-decoder coordinates are plotted for BSC(0.3) with a uniform source, as in Figure 1; see Appendix E for its exact solution. An eigenvalue reaches one (dashed green) precisely at the bifurcation (dashed red vertical), as expected by Conjecture 1 in Section 5.1. In particular, the Jacobian is increasingly ill-conditioned when approaching the bifurcation, as noted in Figure 3 (top). While this allows one to detect the bifurcation, identifying its type is necessary for handling it.

5.1. The IB as a Rate-Distortion Problem

We now explore the intimate relation between the IB and RD, following [5,20]. This leads to a “minimally sufficient” coordinate system for the IB, thereby completing the work of Section 2. In this coordinate system, results [6] on the dynamics of RD roots are readily considered in the IB context. This leads to Conjecture 1, that the IB operator (5) in these coordinates is typically non-singular. The discussion here facilitates the treatment of IB bifurcations in the following Sections 5.2 and 5.3.

First, recall a few definitions. A *rate distortion problem* on a source alphabet \mathcal{X} and a reproduction alphabet $\hat{\mathcal{X}}$ is defined by a *distortion measure* $d : \mathcal{X} \times \hat{\mathcal{X}} \rightarrow \mathbb{R}_{\geq 0}$ (a non-negative function on $\mathcal{X} \times \hat{\mathcal{X}}$ with no further requirements—see Section 2.2 in [28]) and a source distribution $p_X(x)$. One seeks the minimal rate $I(X; \hat{X})$ subject to a constraint D on the expected distortion $\mathbb{E}[d(x, \hat{x})]$ [29,30],

$$R(D) := \min_{p(\hat{x}|x)} \left\{ I(X; \hat{X}) : \mathbb{E}_{p(\hat{x}|x)p_X(x)}[d(x, \hat{x})] \leq D \right\}, \tag{23}$$

known as the *rate-distortion curve*. The minimization is over *test channels* $p(\hat{x}|x)$. A test channel that attains the RD curve (23) is called an *achieving distribution*. We say that an RD problem is *finite* if both of the alphabets \mathcal{X} and $\hat{\mathcal{X}}$ are finite. Using Lagrange multipliers for (23) with $I(X; \hat{X}) + \beta \mathbb{E}[d(x, \hat{x})]$ (normalization omitted for clarity), one obtains a pair of fixed-point equations

$$p(\hat{x}|x) = \frac{p(\hat{x})e^{-\beta d(x, \hat{x})}}{\sum_{\hat{x}} p(\hat{x})e^{-\beta d(x, \hat{x})}} \quad \text{and} \quad p(\hat{x}) = \sum_x p(\hat{x}|x)p(x) \tag{24}$$

in the marginal $p(\hat{x})$ and test channel $p(\hat{x}|x)$, similar to the IB Equations (2) and (4). Iterating over these is Blahut’s algorithm for RD [8], denoted BA_{β}^{RD} here. As with the IB (1), β parameterizes the slope of the optimal curve (23) also for RD. See [28] or [31] for an exposition of rate-distortion theory.

We clarify a definition needed to rewrite the IB as an RD problem. We define the *simplex* $\Delta[S]$ on a (possibly infinite) set S as the collection of finite formal convex combinations $\sum_s a_s \cdot s$ of elements of S . That is, as the S -indexed vectors $(a_s)_{s \in S}$ (equivalently, as functions mapping each s in S to a real number a_s) that satisfy $\sum_s a_s = 1$ and $a_s \geq 0$, with a_s non-zero for only finitely many elements s (the *support* of $(a_s)_s$). Addition and multiplication are defined pointwise, as in $\sum_s a_s \cdot s + \sum_s b_s \cdot s = \sum_s (a_s + b_s) \cdot s$. $\Delta[S]$ is closed under finite convex combinations because the sum of finitely supported vectors is finitely supported. When taking $S = \{e_1, \dots, e_n\}$ the standard basis vectors $(e_i)_j = \delta_{i,j}$ of \mathbb{R}^n , then one can identify the formal operations with those in \mathbb{R}^n , reducing the simplex $\Delta[S]$ to its usual definition. We write r for an element of $\Delta[\mathcal{Y}]$. In particular, an element of $\Delta[\Delta[\mathcal{Y}]]$ is merely a finite convex combination $\sum_{\hat{x}} p(\hat{x})r_{\hat{x}}$ of *distinct* probability distributions $r_{\hat{x}}(y) \in \Delta[\mathcal{Y}]$ on \mathcal{Y} (note that $\Delta[S]$ is a set). When setting $\hat{\mathcal{X}} \subset \Delta[\mathcal{Y}]$ to be a finite subset of distributions, $|\hat{\mathcal{X}}| < \infty$, then $\Delta[\hat{\mathcal{X}}]$ is a special case of the decoder coordinates of Section 2 (unlike $\Delta[\hat{\mathcal{X}}]$ here, the decoder coordinates of Section 2 are *not* required to have their clusters r distinct).

Now, let a finite IB problem be defined by a joint probability distribution $p_{Y|X}$ p_X , as in Section 1. To write it down as an RD problem [5,20], define the *IB distortion measure* by

$$d_{IB}(x, r) := D_{KL} \left[p_{Y|X=x} || r \right], \tag{25}$$

for $x \in \mathcal{X}$, $r \in \Delta[\mathcal{Y}]$, and $p_{Y|X=x} \in \Delta[\mathcal{Y}]$ the conditional probability distribution at $X = x$. The distortion measure d_{IB} (25) and p_X define an RD problem on the continuous reproduction alphabet $\hat{\mathcal{X}} := \Delta[\mathcal{Y}]$. Minimizing the IB Lagrangian \mathcal{L} (in Section 1) is equivalent to minimizing the Lagrangian of this RD problem [20] (Theorem 5). That is, the IB is a rate-distortion problem when considered in these coordinates. IB clusters $r \in \Delta[\mathcal{Y}]$ assume the role of RD reproduction symbols, while an IB root (considered now as an RD root) is equivalently described either by the probabilities of each cluster—namely, by a

point in $\Delta[\Delta[\mathcal{Y}]]$ —or, by a test channel $p(r|x)$. The astute reader might notice that the IB Equations (2) and (4) are then equivalent to RD's fixed-point Equations (24), with the decoder Equation (3) implied by the IB's Markovity. The IB's Y -information $I(Y; \hat{X})$ equals the expected distortion $\mathbb{E}[d_{IB}(x, \hat{x})]$ in (23) up to a constant [20] (Section 5), and so is linear in the test channel $p(r|x)$. Unlike the finite-dimensional coordinate systems of Section 2, this definition of the IB entails no subtleties due to finite dimensionality, such as duplicate clusters (see more below). However, while it allows us to spell out the IB explicitly as an RD problem, handling an infinite reproduction alphabet is difficult for practical purposes. Since no more than $|\mathcal{X}| + 1$ reproduction symbols are needed to write down an IB root [2], this motivates one to consider the IB's *local* behavior, with clusters fixed.

So instead, one may require the reproduction symbols of d_{IB} (25) to be in a list $(r_{\hat{x}})_{\hat{x} \in \hat{\mathcal{X}}}$ indexed by some finite set $\hat{\mathcal{X}}$, with each $r_{\hat{x}}$ in $\Delta[\mathcal{Y}]$ (the elements $r_{\hat{x}_1}, \dots, r_{\hat{x}_T}$ need *not* be distinct a priori). This defines a finite RD problem, for which d_{IB} (25) is merely an $|\mathcal{X}|$ -by- T matrix. Yet, placing identical clusters in the list $(r_{\hat{x}})_{\hat{x}}$ inadvertently introduces degeneracy to the matrix d_{IB} (25), as discussed below. In [5] (Section 6), $(r_{\hat{x}})_{\hat{x}}$ is taken to be the decoders defined by a given encoder $p(\hat{x}|x)$, as in Equation (11) (Section 2). We shall then refer to d_{IB} (25) as *the distortion matrix defined by $p(\hat{x}|x)$* . When $p_{\beta_0}(\hat{x}|x)$ is an optimal IB root then the problem (d_{IB}, p_X) defined by it is called *the tangent RD problem*. Indeed, its RD curve (23) coincides with the IB curve (1) at this point (since an optimal choice of IB clusters is already encoded into d_{IB} (25), then solving the IB boils down to finding the clusters' optimal weights $p(\hat{x})$, which is an RD problem). However, the curves differ outside this point since IB clusters usually vary with β , while the distortion of the tangent problem was defined at $p_{\beta_0}(\hat{x}|x)$ and so is fixed. By definition (1), it follows that the IB curve is the lower envelope of the curves of its tangent RD problems [5] (Corollary 2). We note that a similar construction can also be carried out in inverse encoder coordinates, cf., [2].

Regardless of the formulation used to rewrite the IB as an RD problem, the associated RD problem has an expected distortion $\mathbb{E}[d_{IB}]$ of $I(X; Y) - I(\hat{X}; Y)$ at an IB root (Section 5 in [20] and Lemma 8 in [5]). That is, the IB is a method of lossy compression that strives to preserve the relevant information $I(\hat{X}; Y)$. Due to the Markov condition, information on Y is available only through X . Thus, one may intuitively consider the IB as a lossy compression method of the information on Y that is embedded in X . These intimate relations between the IB and RD suggest that studying bifurcations in either context could be leveraged to understand the other. Bifurcations in finite RD problems are discussed at length in [6] (Section 6). To facilitate the study of IB bifurcations in the sequel (Sections 5.2 and 5.3) using results from RD, we need a “minimally-sufficient” coordinate system for the IB.

Consider an IB root in decoder coordinates as finitely many $p(\hat{x})$ -weighted points $r_{\hat{x}}(y)$ in $\Delta[\mathcal{Y}]$, as in Section 2. Exchanging to decoder coordinates (Equation (11) there) is well-defined as long as there are no zero-mass clusters, $\forall \hat{x} p(\hat{x}) \neq 0$. Yet, even then, the points $r_{\hat{x}}$ in $\Delta[\mathcal{Y}]$ yielded by BA's steps 4 through 6 (Algorithm 1) need *not* be distinct. Namely, they may yield identical clusters $r_{\hat{x}} = r_{\hat{x}'}$ at distinct indices $\hat{x} \neq \hat{x}'$. This leads to a discussion of structural symmetries of the IB (its degeneracies), which is not of use for our purposes; cf., [9]. To avoid such subtleties, we shall say that an IB root is *reduced* if it has no zero-mass clusters, $\forall \hat{x} p(\hat{x}) \neq 0$, and all its clusters are distinct, $\hat{x} = \hat{x}' \Leftrightarrow r_{\hat{x}} = r_{\hat{x}'}$. A root that is not reduced is called *degenerate* or *degenerately represented*. An IB root can be reduced by removing clusters of zero mass and merging identical clusters of distinct indices—see our reduction algorithm in Section 5.2 below. It is straightforward to see from the IB Equations (2)–(4) that reduction preserves the property of being an IB root. Similarly, reducing a root does not change its location in the information plane. So, a root achieves the IB curve (1) if and only if its reduction does. Therefore, reduction decreases the dimension in which the problem is considered while preserving all its essential properties. This allows us to represent an IB root on the smallest number of clusters possible—its *effective cardinality*—by factoring out the IB's structural symmetries. See also [13] (2.3 in Chapter 7), upon which this definition is based.

While the purpose of reduction is to mod-out redundant kernel coordinates (Section 1), it highlights the differences between the various IB definitions found in the literature, bringing to light a subtle caveat of finite dimensionality. To see this, note that reduction could have been defined above in terms of the other coordinate systems of the IB. Its definition in inverse encoder coordinates is nearly identical to that above, while defining it in encoder coordinates is a straightforward exercise. Since the coordinate systems of Section 2 are equivalent at an IB root (without zero-mass clusters), the precise definition does not matter then. Each of these parameterizations encodes the coordinates $r(y)$ of a root's clusters r using a finite-dimensional vector x (note Equation (11)). This enables one to represent duplicate clusters $\hat{x} \neq \hat{x}'$ with $r_{\hat{x}} = r_{\hat{x}'}$, and obliges one to choose the order in which clusters are being encoded into the coordinates of x . A finite-dimensional representation x of an IB root is invariant to interchanging clusters $\hat{x} \neq \hat{x}'$ precisely when they are identical, $r_{\hat{x}} = r_{\hat{x}'}$. The IB's functionals (e.g., its X - and Y -information) are invariant to any cluster permutation; cf., [9,19]. Both of these structural symmetries result from using a finite-dimensional parameterization, with the former eliminated by reduction. In contrast, the elements of $\Delta[\mathcal{Y}]$ are distinct by definition (since $\Delta[\mathcal{Y}]$ is a set), and so parameterizing the IB by points in $\Delta[\Delta[\mathcal{Y}]]$ does not permit identical clusters. An element $\sum_r p(r)r$ of $\Delta[\Delta[\mathcal{Y}]]$ assigns a probability mass $p(r)$ to every point r in $\Delta[\mathcal{Y}]$, with only finitely many points r supported. Thus, it implicitly encodes all the entries $r(y)$ of every probability distribution $r \in \Delta[\mathcal{Y}]$ in a "one size fits all" approach, giving no room for the choices above. This leads us to argue that the IB's structural symmetries are *not* an inherent property but rather an artifact of using its finite-dimensional representations. This is best understood in the context of discontinuous bifurcations, in Section 5.3 below. For comparison, both of the IB formulations [2,20] do not impose an a priori restriction on the number of clusters. The latter does not enable one to encode duplicate clusters, while the former does. The formulation [1] ignores these subtleties altogether, and [9,19] consider the IB on a pre-determined number of possibly duplicate clusters.

In rate-distortion, the *reduction* of a finite RD problem is defined similarly [6] (Section 3.1), by removing a symbol \hat{x} from the reproduction alphabet $\hat{\mathcal{X}}$ and its column $d(\cdot, \hat{x})$ from the distortion matrix once it is not in use anymore (of zero mass). A distortion matrix d is *non-degenerate* if its columns are distinct, $d(\cdot, \hat{x}) \neq d(\cdot, \hat{x}')$ for all $\hat{x} \neq \hat{x}'$. Non-degeneracy arises naturally when considering the RD problem tangent to a given IB root $p(\hat{x}|x)$. Indeed, the distortion matrix d_{IB} (25) defined by $p(\hat{x}|x)$ has duplicate columns if the root has identical clusters, while the other direction holds under mild assumptions (if the $|\mathcal{X}|$ vectors $p_{Y|X=x}$ span $\mathbb{R}^{|\mathcal{Y}|}$, then $D_{KL}[p_{Y|X=x}||r_{\hat{x}}] = D_{KL}[p_{Y|X=x}||r_{\hat{x}'}]$ for all x implies that $r_{\hat{x}} = r_{\hat{x}'}$). Under these assumptions, the distortion matrix induced by an IB root $p(\hat{x}|x)$ is reduced and non-degenerate precisely when $p(\hat{x}|x)$ is a reduced IB root.

Reduction in RD provides the means to show that the dynamics underlying the RD curve (23) are piecewise analytic in β [6], under mild assumptions. Just as in definition (5) of the IB operator, [4] (Equation (5)) similarly define the *RD operator* $Id - BA_{\beta}^{RD}$ in terms of Blahut's algorithm for RD [8]. By using their Theorem 1, [6] (Section 3.1) observed that reducing a finite RD problem to the support of a given RD root mods-out redundant kernel coordinates if the distortion measure is finite and non-degenerate (the *support* of $p(\hat{x})$ is defined by $\text{supp } p(\hat{x}) := \{\hat{x} : p(\hat{x}) > 0\}$). That is, the Jacobian $D(Id - BA_{\beta}^{RD})$ of the RD operator on the reduced problem is then non-singular (in the right coordinate system—see therein), just as with our toy problem (10) in Section 1. By the Implicit Function Theorem, there is therefore a unique RD root of the reduced problem through the given one; this root is real-analytic in β (details there). Considering this for the RD problem tangent to a reduced IB root immediately yields the following:

Corollary 1. *Let $p_{\beta_0}(\hat{x}|x)$ be a reduced IB root of a finite IB problem defined by $p_{Y|X}$ p_X , such that the matrix $p_{Y|X}$ is of rank $|\mathcal{Y}|$. Then, near β_0 , there is a unique function continuous in β , which is a root of the tangent RD problem through $p_{\beta_0}(\hat{x}|x)$; it is real-analytic in β .*

Corollary 1 shows that the local approximation of an IB problem (the roots of its tangent RD problem) is guaranteed to be as well-behaved as one could hope for, provided that the IB is viewed in the right coordinate system. Note, however, that the RD root through $p_{\beta_0}(\hat{x}|x)$ of the tangent problem does *not* in general coincide with the IB root outside of β_0 since the IB distortion d_{IB} (25) varies along with the clusters that define it. However, when the IB clusters are fixed, then one might expect that the Jacobian (13) of BA_β in log-decoder coordinates would be the same as the Jacobian of its RD variant. Indeed, the Jacobian matrix of BA_β^{RD} is the $T \times T$ bottom-right sub-block of the Jacobian (13) of BA_β , up to a multiplicative factor. For this, see Equations (5) and (6) in [4], Equations (14) and (13) in Section 3, and (A25) in Appendix B.2.

As in RD, we argue that reduction in the IB also provides the means to show that the dynamics underlying the optimal curve (1) are piecewise analytic in β . Corollary 1 concludes that, under mild assumptions, through every reduced IB root passes a unique real-analytic RD root. However, its crux is that the Jacobian of the RD operator $Id - BA_\beta^{RD}$ is non-singular at a reduced root. Due to the IB's close relations with RD, and since reduction in the IB is a natural extension of reduction in RD, we argue that the same is also to be expected of the IB operator $Id - BA_\beta$ (5) in decoder coordinates. To see this, note that IB roots are finitely supported [2] (Lemma 2.2(i)), and so one may take finitely supported probability distributions $\Delta[\Delta[\mathcal{Y}]]$ for the IB's optimization variable. Thus, the IB's BA_β operator in decoder coordinates (of Section 2) may be considered as an operator on $\Delta[\Delta[\mathcal{Y}]]$. Next, consider the RD problem defined by p_X and d_{IB} (25) on the continuous reproduction alphabet $\Delta[\mathcal{Y}]$, as in [20]. This defines on $\Delta[\Delta[\mathcal{Y}]]$ also the BA operator BA_β^{RD} for RD. Now that both BA operators are considered on an equal footing, we note the following. First, while BA_β^{RD} iterates over the IB Equations (2) and (4), its IB variant BA_β iterates also over the decoder Equation (3) (plug the IB distortion measure d_{IB} (25) into the Equations (24) defining BA_β^{RD} to see this). The latter Equation (3) is a necessary condition for $Y \rightarrow X \rightarrow \hat{X}$ to be Markov, and so can be understood as an enforcement of Markovity (in contrast, an arbitrary triplet (Y, X, \hat{X}) of random variables only satisfies $p(y|\hat{x}) = \sum_x p(y|x, \hat{x})p(x|\hat{x})$). That is, IB roots are RD roots with an extra constraint. Second, by Theorem 1 in [4], reducing $Id - BA_\beta^{RD}$ from the continuous reproduction alphabet $\Delta[\mathcal{Y}]$ to a root of finite support renders it non-singular, under mild assumptions. This suggests that reducing $Id - BA_\beta$ (5) from $\Delta[\Delta[\mathcal{Y}]]$ to a root's effective cardinality should also render it non-singular, due to the similarity between these operators, and since reduction in the IB is a natural extension of reduction in RD. In line with the discussion of Section 1 on reduction, we therefore state the following:

Conjecture 1. *The Jacobian matrix $I - D_{\log p(y|\hat{x}), \log p(\hat{x})} BA_\beta$ at (16) of the IB operator (5) in log-decoder coordinates is non-singular at reduced IB roots so long as it is well-defined, except perhaps at points of bifurcation.*

The intuition behind this conjecture stems from analyticity, as follows. The IB operator $Id - BA_\beta$ (5) is real-analytic, since each of the Equations 1.4–1.8 defining it (in the BA-IB Algorithm 1) is real-analytic in its variables. For a root x_0 of a real-analytic operator F , one might expect that, in general, (i) no roots other than x_0 exist in its vicinity and that (ii) $D_x F|_{x_0}$ has no kernel. That is, unless the operator is degenerate at x_0 in some manner or x_0 is a bifurcation. To see this, recall [32] (Section IX.3) that a real-valued function F_i in $x \in \mathbb{R}^n$ is *real-analytic* in some open neighborhood of x_0 if it is a power series in $x = (x_1, \dots, x_n)$, within some radius of convergence (although a strictly positive radius is needed, we omit these details for clarity). For every practical purpose, one may replace F_i by a polynomial in (x_1, \dots, x_n) when x is close enough to the base-point x_0 , by truncating the power series. Viewed this way, a root of an operator $F(x) = (F_1(x), \dots, F_n(x))$ is nothing but a solution of n polynomial equations in n variables. However, a square polynomial system typically has only isolated roots, which is (i). This is best understood in terms of Bézout's Theorem; see [33] (6 in IV.4) for example. For (ii), a vector v is in $\ker D_x F$

precisely when it is orthogonal to each of the gradients ∇F_i . However, ∇F_i is the vector of the first-order monomial coefficients of x_1, \dots, x_n in F_i . In a general position, these n coefficient vectors $\nabla F_1, \dots, \nabla F_n$ are linearly independent, and so v must vanish as claimed. If F is degenerate such that $F_i = F_j$ for particular $i \neq j$, for example, then both points fail, of course. See also Section I.2 of [34] for (i) and (ii). This intuition accords with the comments of [28] (Section 2.4) on RD: “usually, each point on the rate distortion curve [...] is achieved by a unique conditional probability assignment. However, if the distortion matrix exhibits certain form of symmetry and degeneracy, there can be many choices of [a minimizer]”. Indeed, the fact that the dynamics underlying the RD curve (23) are piecewise real-analytic [6] (under mild assumptions) can be similarly understood to stem from the analyticity of the RD operator $Id - BA_\beta^{RD}$.

Subject to Conjecture 1, a Jacobian eigenvalue of the IB operator (5) must vanish gradually as one approaches a bifurcation, causing the critical slowing down of BA-IB [4] (observe that BA’s Jacobian (13) is continuous in the root at which it is evaluated). When an IB root traverses a bifurcation in which its effective cardinality decreases, then it is not reduced anymore. One can then handle the bifurcation by reducing the root anew. To ensure proper handling by the bifurcation’s type, we consider the latter closely in Sections 5.2 and 5.3 below. In a nutshell, following the IB’s ODE (16) along with a proper handling of its bifurcations is the idea behind our root-tracking algorithm (in Section 6), for approximating the IB numerically.

Conjecture 1 is compatible with our numerical experience. However, we leave its proof to future work. To that end, one could examine closely the smaller matrix S (17) (of Lemma 1 in Section 3), for example. However, even if Conjecture 1 were violated, then one could detect that easily by inspecting the Jacobian’s eigenvalues. Conjecture 1 also implies that IB roots are locally unique outside of bifurcations when presented in their reduced form. Non-uniqueness of optimal roots is detectable by inspecting the Jacobian’s eigenvalues—see Corollary 3 in Section 5.3 and the discussion following it. See also Section 6.3 in [6] for the respective discussion in RD. With that, most of the results in Sections 5.2 and 5.3 below do not depend on the validity of Conjecture 1.

5.2. Continuous IB Bifurcations: Cluster Vanishing and Cluster Merging

Following [10], we consider the evolution of IB roots which are a continuous function of β . By representing an IB root in its reduced form (Section 5.1), it is evident that there are two types of *continuous* IB bifurcations. We provide a practical heuristic (Algorithm 2) for identifying and handling such bifurcations. The discussion here is complemented by Section 5.3 below, which considers the case where continuity does not hold.

The evolution of an IB root in β obeys the ODE (16) as long as it can be written as a differentiable function in β , as in Theorem 1. Considering the root in decoder coordinates, this amounts to an evolution of a T -tuple of points $r_{\hat{x}}$ in $\Delta[\mathcal{Y}]$ and their weights $p(\hat{x})$. These typically traverse the simplex smoothly as the constraint β is varied, as demonstrated in Figure 6. We now consider two cases where this evolution does not obey the ODE (16), due to violating differentiability.

Consider an optimal IB root in its reduced form (see Section 5.1). Namely, consider the reduced form of a root that achieves the IB curve (1). Suppose that its decoders $r_{\hat{x}}$ and weights $p(\hat{x})$ are continuous in β . Then, a qualitative change in the root can occur only if either (i) two (or more) of its clusters collide or (ii) the marginal probability $p(\hat{x})$ of a cluster \hat{x} vanishes. In either case, the minimal number of points in $\Delta[\mathcal{Y}]$ required to represent the root decreases. That is, its effective cardinality decreases (a qualitative change where the effective cardinality increases is obtained by merely reversing the dynamics in β). We call the first a *cluster-merging bifurcation* and the second a *cluster-vanishing bifurcation*, or *continuous bifurcations* collectively. Both types were observed already in [17] (Section IV.C) in the related setting of RD problems with a continuous source alphabet. Among the two, cluster-vanishing bifurcations are more frequent in practice than cluster merging. This can

be understood by considering cluster trajectories in the simplex. In a general position, one might expect clusters to seldom be at the same “time” and place (that is, β and $r \in \Delta[\mathcal{Y}]$).

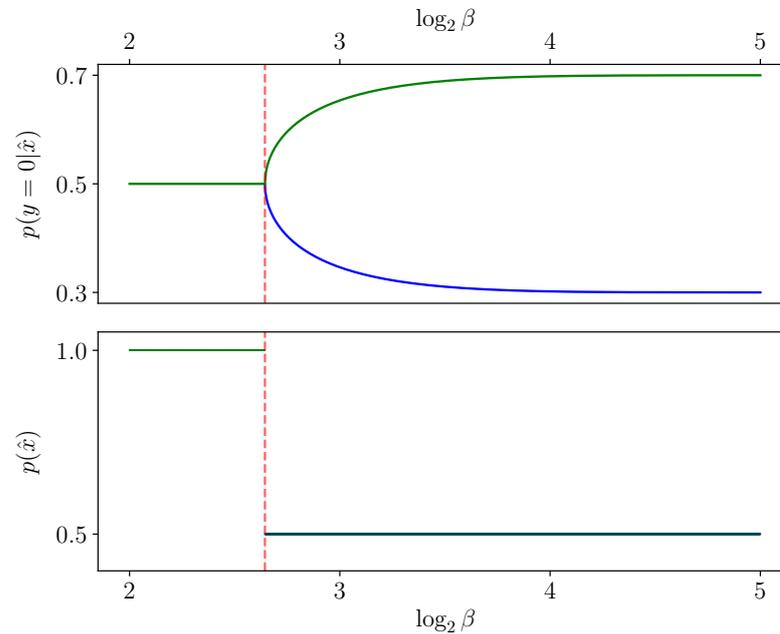


Figure 6. A cluster-merging bifurcation. The reduced form of the optimal IB root in decoder coordinates as a function of β , for the exact solution of BSC(0.3) with a uniform source, as in Figure 1 (see Appendix E). At high enough β , the root consists of two clusters (in green and blue), each of a marginal probability $\frac{1}{2}$. The clusters collide at $\beta_c = 61/4$ (dashed red vertical) and merge to one, yielding the trivial solution—a single cluster of probability 1 at p_Y . Carefully note that only a *single* IB root is plotted here, in its reduced form, with one cluster to the left of β_c and two to the right. The violation of clusters’ differentiability at β_c can be observed visually (**top**), and the root is otherwise real-analytic in β , as can be deduced from Figure 5. Since the trivial solution is an IB root for every $\beta > 0$ (not shown), then β_c is indeed a bifurcation, where the trivial and non-trivial roots intersect. To see this, consider the degenerate form of the trivial solution on *two* copies of p_Y , each of probability $\frac{1}{2}$. The marginals $p(x)$ (**bottom**) appear to be discontinuous at β_c because the root was reduced before plotted (the latter degenerate form of the trivial root is *not* plotted to the left of β_c).

We argue that cluster merging and cluster vanishing are indeed bifurcations, where IB roots of distinct effective cardinalities collide and merge into one. We offer two ways to see this. First, using the inverse encoder formulation of the IB in [2] (Section II.A), one can consider an optimization problem in which the number of IB clusters is constrained explicitly (the inverse encoders of an IB root with no zero-mass clusters are in bijective correspondence with its decoders, as noted in Section 2, and so inverse encoder and decoder coordinates are interchangeable). By the arguments therein, the constrained problem has an optimal root (due to compactness), which achieves the optimal curve of the constrained problem. The latter curve must be sub-optimal if fewer clusters are allowed than needed to achieve the IB curve (1). Thus, whenever the effective cardinality of an optimal root (in the un-constrained problem) decreases, it must therefore collide with an optimal root of the constrained IB problem (by Corollary 3 in Section 5.3 below). This accords with [1] (Section 3.4), which describes IB bifurcations as a separation of optimal and sub-optimal IB curves according to their effective cardinalities. Second, consider the reduced form of an IB root at the point of a continuous bifurcation. Since its effective cardinality decreases there strictly, say from T_2 to T_1 , then the root can be represented on T_1 clusters at the bifurcation itself. However, the Jacobian of the IB operator (5) in log-decoder coordinates is non-singular when represented on T_1 clusters, as discussed after Proposition 1 (in Section 5.3). Thus, by the Implicit Function’s Theorem, there is a unique IB root on T_1 clusters through

this point. It exists at both sides of the bifurcation (above and below the critical point). When represented on T_2 clusters, however, the latter intersects at the bifurcation with the root of effective cardinality T_2 , and so the two roots collide and merge there to one. This argument is identical to [6] (Section 6.2), which proves that distinct RD roots collide and merge at cluster-vanishing bifurcations in RD.

At a continuous bifurcation, IB roots of distinct effective cardinalities collide and merge into one, as discussed above. Specifically, one root achieves the minimal value of the IB Lagrangian and so is stable, while the other root is sub-optimal. As we shall now elaborate, continuous IB bifurcations are thus *pitchfork bifurcations* (e.g., Section 3.4 in [35]), in accordance with [19]. Even though the optimal root is continuous in β (by assumption), its differentiability is violated at the point of bifurcation. This can be inferred from the comments following Theorem 1 and seen in Figure 6. Strictly speaking, several copies of the root of larger effective cardinality collide at a continuous bifurcation. When two clusters $r \neq r'$ collide in a cluster merging bifurcation, then the root itself is invariant to interchanging their coordinates after the collision but not before it, breaking the IB's first structural symmetry discussed in Section 5.1. Interchanging the coordinates of r and r' (and their marginals) before the collision yields two distinct copies of essentially the same root. For a cluster vanishing bifurcation, the IB's functionals (e.g., its X - and Y -information) do not depend on the coordinates $(r(y))_y$ of a vanished cluster r , rendering these redundant; cf., [9] (Section 3.1). Before the cluster r vanishes, there is one copy of the root for each index \hat{x} , with r placed at its \hat{x} coordinates. Considered in reduced coordinates, these coincide to a single copy after the cluster vanishes. This breaks the IB's second structural symmetry.

With that, we note that cluster-vanishing bifurcations *cannot* be detected directly by standard local techniques (i.e., considering the derivative's kernel directions at the bifurcation point), whether considering the Hessian of the IB's loss function as in [9] or the Jacobian of the IB operator (5) as here. The technical reason for this is as follows, while the root cause underlying it is best understood in the context of discontinuous bifurcations (after Proposition 1 in Section 5.3). Observe that the $I(Y; \hat{X})$ and $I(X; \hat{X})$ functionals do not depend on the coordinates $(r(y))_y$ of clusters r of zero mass. Thus, the directions corresponding to these coordinates are always in the kernel regardless of whether evaluating at a bifurcation or not, and so cannot be used to detect a bifurcation (the direction corresponding to a cluster's marginal is useless when one does not know which coordinates $(r(y))_y$ to pick for r). Indeed, with its dynamics in β reversed, "*a new symbol grows continuously from zero mass*" in a cluster-vanishing bifurcation, as [17] (Section IV.C) comments in a related setting. It is then not clear a priori which point in $\Delta[\mathcal{Y}]$ should be chosen for the new symbol, rendering the perturbative condition at Equation (9) difficult to test. In accordance with this, Ref. [9] (Section 5) offers a perturbative condition for detecting arbitrary IB bifurcations, while ref. [13] (3.2 in Part III) offers a condition for detecting cluster-merging bifurcations by analyzing cluster stability. However, both conditions are equivalent (Appendix F), and so must detect the same type of bifurcations. In contrast, a cluster-splitting (or merging) bifurcation is straightforward to detect because the stability of a particular cluster \hat{x} is a property of the root itself—see Appendix F and the references therein for details.

One may wonder whether bifurcations exist in the IB for the same reason as they do in RD. As in the IB, RD problems typically have many sub-optimal curves [6] (Section 6.1). While (continuous) bifurcations in the IB stem from restricting the effective cardinality [1] (Section 3.4), in RD they stem from the various restrictions that a reproduction alphabet has. For example, a reproduction alphabet $\hat{\mathcal{X}} := \{r_1, r_2, r_3\}$ of an RD problem may be restricted to the distinct subsets $\{r_1, r_2\}$ and $\{r_2, r_3\}$, usually yielding distinct sub-optimal RD curves (e.g., Figure 6.1 in [6]). In contrast to RD, the IB's distortion d_{IB} (25) defined by a root's clusters is determined a posteriori by the problem's solution rather than a priori by the problem's definition. As a result, both reasons for the existence of bifurcations coincide. To see this, consider the IB as an RD problem whose reproduction symbols $\hat{\mathcal{X}}$ are a finite subset of $\Delta[\mathcal{Y}]$ which is allowed to vary (i.e., as if defining the tangent RD problem anew at each β). Distinct restrictions of a reproduction alphabet $\hat{\mathcal{X}}$ can be forced to agree by altering

the symbols themselves, so long as they are of the same size. For example, restricting the set $\{r_1, r_2, r_3\}$ of reproduction symbols to $\{r_1, r_2\}$ is the same as restricting it to $\{r_2, r_3\}$ instead, and then replacing r_3 with $r_1 \in \Delta[\mathcal{Y}]$ in the restricted problem (this is not to be confused with cluster permutations, which change the order in which clusters are listed but do *not* alter the symbols themselves).

The dynamical point of view above, considering an IB root as weighted points traversing $\Delta[\mathcal{Y}]$, offers a straightforward way to identify and handle continuous IB bifurcations. It is spelled out as our root-reduction Algorithm 2. For cluster-vanishing bifurcations, one can set a small threshold value $\delta_1 > 0$ and consider the cluster \hat{x} as vanished if $p(\hat{x}) < \delta_1$ (Step 2.3), as in [6] (Section 3.1). Similarly, for cluster-merging bifurcations, one can set a small threshold $\delta_2 > 0$ and consider the clusters $\hat{x} \neq \hat{x}'$ to have merged if $\|r_{\hat{x}} - r_{\hat{x}'}\|_\infty < \delta_2$ (Step 2.9). A vanished cluster is then erased (and merged clusters replaced by one), resulting in an approximate IB root on fewer clusters. This not only identifies continuous IB bifurcations but also handles them, since the output of the root-reduction Algorithm 2 is a numerically reduced root, represented in its effective cardinality. To re-gain accuracy, we shall later invoke the BA-IB Algorithm 1 on the reduced root, as part of our root-tracking algorithm (in Section 6). We note that one should pick the thresholds δ_1 and δ_2 small enough to avoid false detections, and yet not too small so as to cause mis-detections. Mis-detections will be handled later, in Section 6.1, using a heuristic algorithm.

Algorithm 2 Root reduction for the IB

```

1: function REDUCE ROOT( $p(y|\hat{x}), p(\hat{x}); \delta_1, \delta_2$ )
Input:
  An approximate IB root  $(p(y|\hat{x}), p(\hat{x}))$  in decoder coordinates,
  a cluster-mass threshold  $0 < \delta_1 < 1$  and a cluster-merging threshold  $0 < \delta_2 < 1$ .
Output: An approximate IB root  $(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x}))$  at its effective cardinality.
2:   for  $\hat{x}$  do
3:     if  $p(\hat{x}) < \delta_1$  then                                     ▷ Delete clusters of near-zero mass.
4:       delete the coordinates of  $\hat{x}$ , from  $p(\hat{x})$  and  $p(y|\hat{x})$ .
5:     end if
6:   end for
7:    $p(\hat{x}) \leftarrow$  normalize  $p(\hat{x})$    ▷ Preserve normalization, in case clusters were removed.

8:   for  $\hat{x} \neq \hat{x}'$  do
9:     if  $\|p(y|\hat{x}) - p(y|\hat{x}')\|_\infty < \delta_2$  then                 ▷ Merge nearly identical points in  $\Delta[\mathcal{Y}]$ .
10:       $p(\hat{x}) \leftarrow p(\hat{x}) + p(\hat{x}')$ 
11:      delete the coordinates of  $\hat{x}'$ , from  $p(\hat{x})$  and  $p(y|\hat{x})$ .
12:    end if
13:  end for

14:  return  $(p(y|\hat{x}), p(\hat{x}))$ 
15: end function

```

Using the root-reduction Algorithm 2 allows one to stop early in the vicinity of a bifurcation when following the path of an IB root. As mentioned in Section 4, early stopping restricts the computational difficulty of root tracking [6]. Further, reducing the root *before* invoking BA-IB (Algorithm 1) allows us to avoid BA's critical slowing down [4], since reduction removes the nearly vanished Jacobian eigenvalues that pertain to the nearly vanished (or nearly merged) cluster(s), which are the cause of BA's critical slowing down. cf., Proposition 1 (Section 5.3) and the discussion around it. See also [6] (Figure 3.1(C) and Section 3.2) for the respective behavior in RD. Finally, we comment that the root-reduction Algorithm 2 can also be implemented in the other two coordinate systems of Section 2.

5.3. Discontinuous IB Bifurcations and Linear Curve Segments

In the previous Section 5.2, we considered continuous IB bifurcations—namely, when the clusters $r_{\hat{x}} \in \Delta[\mathcal{Y}]$ and weights $p(\hat{x})$ of an IB root are continuous functions of β . By exploiting the intimate relations between the IB and RD (Section 5.1), we now consider IB bifurcations where these *cannot* be written as a continuous function of β . In our experience, discontinuous bifurcations are infrequent in practice. However, the theory they evoke has several subtle consequences of practical implications important for computing IB roots (in Section 6). Though, perhaps more importantly, they oblige one to ask what *is* the IB? We start with several examples before diving into the theory; e.g., Figure 7.

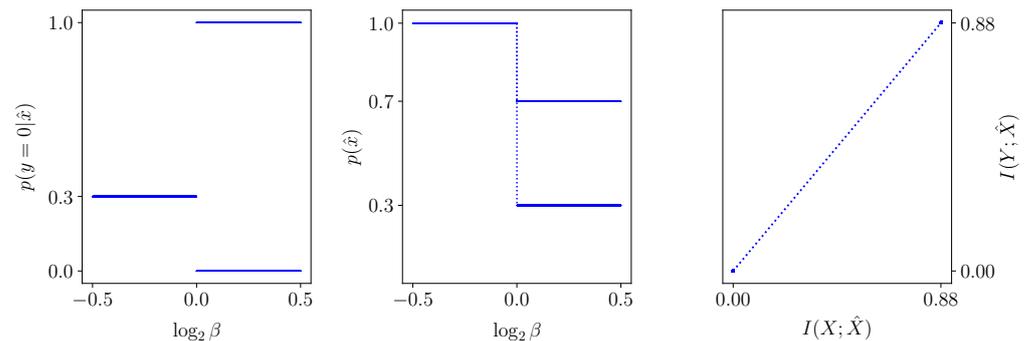


Figure 7. A discontinuous IB bifurcation at $\beta_c = 1$, of the problem defined by $p_{Y|X} p_X = \begin{pmatrix} 0.3 & \\ & 0.7 \end{pmatrix}$. (Left): to the left of β_c , the optimal solution is the trivial one, supported on the IB cluster p_Y . To the right it is supported on the boundary points (1, 0) and (0, 1) of $\Delta[\mathcal{Y}]$. (Middle): the marginals are constant, except at the point of bifurcation. Any convex combination of the trivial and non-trivial roots is optimal there (dotted). That is, this is a **support-switching bifurcation** as in RD [6] (Figure 6.2). (Right): the IB curve exhibits a linear segment of slope $1/\beta_c = 1$, connecting the image of the trivial solution in the information plane (bottom-left) to that of the non-trivial one (top-right). See comments in the main text.

The examples of discontinuous IB bifurcations of which we are aware can be understood in RD context as follows. Consider the IB as an RD problem on the continuous reproduction alphabet $\Delta[\mathcal{Y}]$, with IB roots parameterized by points in $\Delta[\Delta[\mathcal{Y}]]$ (see Section 5.1). In RD, the existence of linear curve segments is well-known [28]—e.g., Figure 2.7.6 in the latter and its reproduction in [6] (Figure 6.2). Section 6.5 in [6] offers an explanation of linear segments in terms of a *support-switching bifurcation*. Namely, a bifurcation where two RD roots of distinct supports exchange optimality at a particular multiplier value β_c . Both roots evolve smoothly in β while only exchanging optimality at the bifurcation. At β_c itself, every convex combination of these two roots is also an RD root. In particular, the optimal RD root *cannot* be written as a continuous function of β . The sudden emergence of an entire segment of roots at β_c can be understood by RD’s convexity and analyticity properties, as follows. The RD curve (23) is parameterized by the slope $-\beta$ of its tangents [28] (Theorems 2.5.1 and 2.5.2). Above and below β_c , specifying the tangent’s slope determines a curve-achieving distribution on the optimal root (the root whose curve is lower at this slope value). Equivalently, the lower convex envelope of these roots in the RD plane coincides with one root above β_c and with the other below it, as seen in Figure 8 (black). At β_c itself, specifying the slope determines a distribution on both roots. Thus, the convexity of the RD curve and of the set of achieving distributions implies a linear segment at β_c (Theorem 2.4.1 in [28] and Theorem 5 below). Finally, this behavior is possible due to analyticity, since the roots of a real-analytic operator $Id - BA_\beta^{RD}$ are either isolated (typical) or an algebraic curve (atypical) by Bézout’s Theorem—see (i) in the discussion following Conjecture 1.

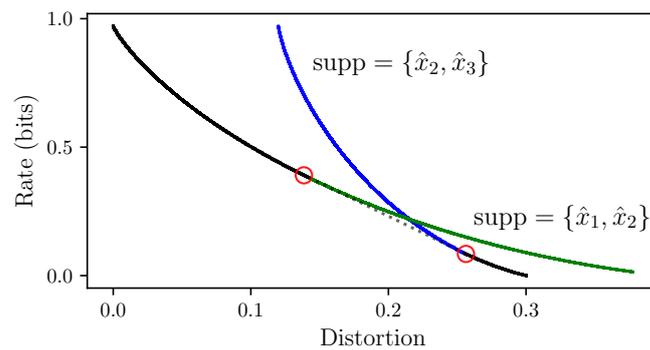


Figure 8. A support-switching bifurcation in RD, reproducing Figure 6.2(F) in [6] (details therein). The RD curve (23) (black) is the envelope of its tangents, parameterized by their slope $-\beta$, [28]. At high slopes, the envelope coincides with that of the problem restricted to the reproduction alphabet $\{\hat{x}_1, \hat{x}_2\}$ (green), and at low slopes with that restricted to $\{\hat{x}_2, \hat{x}_3\}$ (blue). At a critical slope $-\beta_c$, the tangent touches both curves (red circles). Convexity then implies a linear segment (dashed)—see main text.

For one example of linear curve segments in the IB, say that a matrix M decomposes if it can be written (non-trivially) as a block matrix by permuting its rows or columns. In light of the above, we have the following refinement of Theorem 2.6 in [2]:

Theorem 3. *The IB curve (1) has a linear segment at $\beta = 1$ if and only if the problem’s definition $p_{Y|X} p_X$ decomposes.*

Recall that the slope of the IB curve is $1/\beta$ at a multiplier value β [1] (Equation (32)). Thus, Theorem 3 equates decomposable problems with linear curve segments of slope 1 (the slope cannot exceed one due to the data processing inequality). Figure 7 provides a simple decomposable example, exhibiting a support-switching bifurcation between its trivial and non-trivial roots. Non-decomposable examples also exist, exhibiting a support-switching bifurcation at lower slope values (higher critical β ’s). For example, a symmetric binary erasure channel exhibits a support-switching bifurcation [2] (Section IV.B), which is manifested by a linear segment of slope $1/\beta_c \leq 1$, for $\beta_c \geq 1$ (switching between the trivial root at p_Y and a bi-clustered root supported on $(\beta_c^{-1}, 1 - \beta_c^{-1}, 0)$ and $(0, 1 - \beta_c^{-1}, \beta_c^{-1}) \in \Delta[\mathcal{Y}]$; the linear segment of slope β_c^{-1} is Equation (4.8) there). See [2] (Section IV) for further examples. We argue that in the IB, support-switching bifurcations exhibit the same behavior as in RD. That is, two roots that evolve smoothly in β and exchange optimality at the bifurcation. While the sequel can justify this in general, there is a simple way to see this in practice. Namely, following the two roots of Figure 7 through the bifurcation by using BA-IB with deterministic annealing [11] (follow the trivial root of Figure 7 from left to right and the non-trivial one from right to left, through the bifurcation at $\beta_c = 1$ there). As deterministic annealing usually follows a solution branch continuously, this immediately reveals either root at the region where it is sub-optimal (not displayed).

A support-switching bifurcation evidently has similar characteristics to a *transcritical* bifurcation (e.g., Section 3.2 in [35]), though it should perhaps be classified as an *imperfect transcritical* since the roots do not intersect per se as in a classical transcritical. This extends the results of [19], who conclude that IB bifurcations “are only of pitchfork type” (Theorem 5 therein says that the bifurcations detected by their Theorem 3 are degenerate rather than transcritical, concluding that “the bifurcation guaranteed by Theorem 3 is [generically] pitchfork-like”). To see the reason for this discrepancy, note that they employ the mathematical machinery in [36] of bifurcations under symmetry. Since pitchfork bifurcations are “common in physical problems that have a symmetry” [35] (Section 3.4), then detecting only pitchforks by using the above machinery might not come as a surprise. Both [9] and its sequel [19] consider the IB’s symmetry to interchanging the coordinates of identical clusters (Definition

1(1) in [19]). However, this is a structural symmetry of the IB which stems from representing IB roots by finite-dimensional vectors (Section 5.1), and is broken in continuous IB bifurcations (Section 5.2). On the other hand, discontinuous IB bifurcations need not break this symmetry, as can be seen by inspecting the roots of Figure 7 closely (the trivial solution to the left of β_c there may be given a degenerate bi-clustered representation, which is fully supported on p_Y but has a second cluster $r \neq p_Y$ of zero mass. Neither of its roots then possesses a symmetry to interchanging cluster coordinates, at either side of β_c).

A few convexity results from rate-distortion theory are needed to consider discontinuous bifurcations in general. These have subtle practical implications, which are of interest in their own right.

Theorem 4 (Theorem 2.4.2 in [28]). *The set of conditional probability distributions $p(\hat{x}|x)$ which achieve a point $(D, R(D))$ on the rate-distortion curve (23) is convex.*

Viewing the IB as an RD problem as in [20] immediately yields an identical result for the IB:

Corollary 2. *The set of IB encoders that achieve a point (I_X, I_Y) on the IB curve (1) is convex.*

The proof is provided below for completeness. We note that a version of Corollary 2 in inverse encoder coordinates can also be synthesized from the ideas leading to Theorem 2.3 in [2].

Proof of Corollary 2. Consider a finite IB problem $p_{Y|X} p_X$ as an RD problem (d_{IB}, p_X) on the continuous reproduction alphabet $\Delta[\mathcal{Y}]$, as defined by (25) in Section 5.1. As noted above, its encoders (or test channels) are conditional probability distributions $p(r|x)$, with $r \in \Delta[\mathcal{Y}]$, supported on finitely many coordinates (r, x) .

Let $p_1(r|x)$ and $p_2(r|x)$ be encoders achieving a point (I_X, I_Y) on the IB curve (1). Define their support by $\text{supp } p(r|x) := \text{supp } p(r)$, where $p(r)$ is defined from $p(r|x)$ via marginalization, as in (4). By Theorem 5 in [20], $p_1(r|x)$ and $p_2(r|x)$ may be considered as test channels achieving the curve (23) of the RD problem (d_{IB}, p_X) . The reproduction symbols $r \in \Delta[\mathcal{Y}]$ supporting a convex combination $p_\lambda := \lambda \cdot p_1 + (1 - \lambda) \cdot p_2$, $0 \leq \lambda \leq 1$, are contained in the supports of p_1 and p_2 : $\text{supp } p_\lambda \subseteq \text{supp } p_1 \cup \text{supp } p_2$. Therefore, p_λ is finitely supported. Although Berger's Theorem 4 assumes that the reproduction alphabet is finite, one can readily see that its proof works just as well when the distributions involved are finitely supported. Thus, by Theorem 4, p_λ achieves the above point on the RD curve (23). Since this point (I_X, I_Y) is on the IB curve (1), then p_λ is an optimal IB root. \square

The RD curve (23) is the envelope of lines of slope $-\beta$ and intercept $\min_{p(\hat{x}|x)} (I(X; \hat{X}) + \beta \mathbb{E}[d(x, \hat{x})])$ along the R -axis, e.g., [28]. Thus, Theorem 4 can be generalized by considering the achieving distributions that pertain to a particular slope value rather than to a particular curve point $(D, R(D))$ —see [6] (Section 6.3).

Theorem 5 (Theorem 20 in [6]). *For any $\beta > 0$ value, the set of distributions achieving the RD curve (23) that correspond to β is convex.*

As with Corollary 2, we immediately have an identical result for roots achieving the IB curve (1):

Corollary 3. *For any $\beta > 0$ value, the set of optimal IB encoders that correspond to β is convex.*

See also [2] (Section IV) for an argument in inverse encoder coordinates. In particular, note the duality technique leading to (b) and (c) in Theorem 4.1 there. This duality boils down to describing a compact convex set in the plane by its lines of support, as in the observation leading to Theorem 5. Commensurate with the IB being a special case of RD,

Corollary 3 can also be proven directly from the IB's definitions in direct encoder terms [37]. Note that the requirement that the IB root indeed achieves the curve is necessary. Otherwise, one could take convex combinations with the trivial IB root $p(r|x) = \delta_{r,p_Y}$ (which satisfies the IB Equations (2)–(4) for every $\beta > 0$, as one can verify directly). This yields absurd results, since the trivial root contains no information on either X or Y .

As in [6] (Section 6.3), the convexity of optimal IB roots (Corollary 3) has several important consequences. For one, unlike the (local) bifurcations we have considered so far, bifurcation theory also has *global bifurcations*. These are “bifurcations that cannot be detected by looking at small neighborhoods of fixed points” [12] (Section 2.3). From convexity, it immediately follows that

Corollary 4. *There are no global bifurcations in finite IB problems.*

Indeed, if at a given β value there exists more than one optimal root, then the Jacobian of the IB operator $Id - BA_\beta$ (5) must have a kernel vector pointing along the line connecting these optimal roots, by Corollary 3.

With that comes an important practical caveat. Corollaries 2 and 3 hold for the IB when parameterized by points in $\Delta[\Delta[\mathcal{Y}]]$. However, the above kernel vector (which exists due to convexity) may not be detectable if an IB root is improperly represented by a finite-dimensional vector. For example, consider the bifurcation in Figure 7, where a line segment at β_c connects the trivial (single-clustered) root to the 2-clustered root. Obviously, the bifurcation there cannot be detected by the Jacobian of the IB operator (5) when it is computed on $T = 1$ clusters (Jacobian of order $1 \cdot (|\mathcal{Y}| + 1)$). Indeed, the root of effective cardinality two cannot be represented on a single cluster, and so the line segment connecting it to the trivial root does not exist in a 1-clustered representation. This is demonstrated in Figure 9, which compares Jacobian eigenvalues at reduced representations to those at 2-clustered representations. The same reasoning gives the following necessary condition:

Proposition 1 (A necessary condition for detectability of IB bifurcations). *A bifurcation at β_c in a finite IB problem which involves roots of effective cardinalities T_1 and T_2 is detectable by a non-zero vector in $\ker(I - D_{\log p(y|\hat{x}), \log p(\hat{x})} BA_{\beta_c})$ only if the latter is evaluated at a representation on at least $\max\{T_1, T_2\}$ clusters.*

Indeed, suppose that $T_1 \not\geq T_2$ (the conclusion is trivial if $T_1 = T_2$). By definition, a root of effective cardinality T_2 does not exist in representations with less than T_2 clusters. Thus, there is no bifurcation in a T -clustered representation if $T < T_2$, and so there is then nothing to detect. As a special case of this argument, note that Conjecture 1 (Section 5.1) implies that the Jacobian is non-singular in a T_1 -clustered representation of the T_1 -clustered root (namely, at its reduced representation). With that, we have observed numerically that the eigenvalues of $D_{\log p(y|\hat{x}), \log p(\hat{x})} BA_\beta$ do *not* depend on the representation's dimension if computed on strictly more clusters than the effective cardinality (which makes sense considering Theorem 2.6.1 in [28] or Lemma 2.2(i) in [2]). Rather, only the eigenvalues' multiplicities vary by dimension. We omit practical caveats on exchanging between the coordinate systems of Section 2 for brevity.

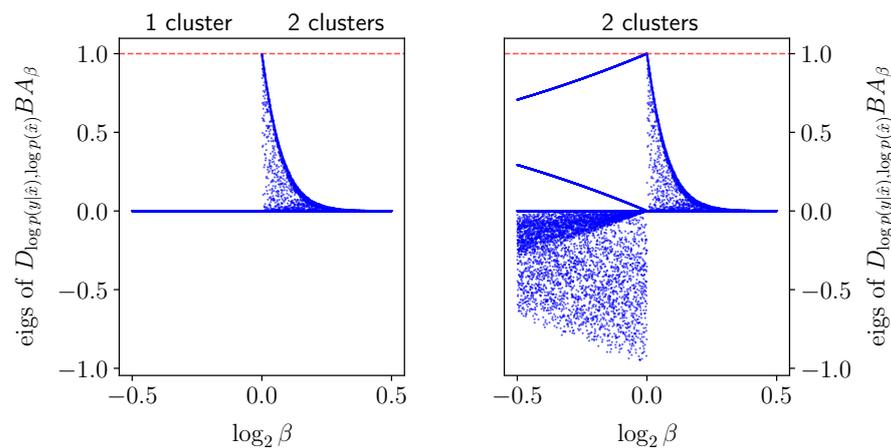


Figure 9. Bifurcations can be detected by BA_β 's Jacobian only if computed on enough clusters. The approximate eigenvalues of $D_{\log p(y|x), \log p(x)} BA_\beta$ are plotted by the representation's dimension for the problem in Figure 7. The eigenvalues are evaluated at solutions obtained by the BA-IB Algorithm 1 (stopping condition = 10^{-9}), initialized anew at random for each β . While the random initializations account for much of the eigenvalues' spread, they reveal the solution's behavior through its various approximations. Other factors which contribute to this spread are the degeneracy of the solutions (when $\beta < 1$, right panel), BA's loss of accuracy near the bifurcation (Figure 3 bottom), and the decoders' proximity to the simplex boundaries (see Equation (13)). (Left): when computed at reduced representations (on $T = 1$ clusters to the left, $T = 2$ to the right), then the eigenvalues at the trivial solution give no indication of the upcoming bifurcation (at $\beta < 1$), unlike the eigenvalues at the 2-clustered root ($\beta > 1$). (Right): the bifurcation's presence is clearly noticed also at the trivial solution ($\beta < 1$) when evaluated at its degenerate 2-clustered representations. Indeed, the trivial solution is then represented on the same number of clusters ($T = 2$) as the root to the right ($\beta > 1$)—see Proposition 1. However, due to the bifurcation, the eigenvalues' trajectories are not smooth at $\beta_c = 1$. Both: a similar dependency on the representation's dimension also exists in the other bifurcation examples in this paper (though without the eigenvalues' spread).

The discussion of discontinuous bifurcations naturally leads one to consider the IB as an RD problem on the continuous reproduction alphabet $\Delta[\mathcal{Y}]$, as in Corollaries 2 and 3, unlike its usual definitions in the literature. When considered this way, IB roots are merely paths $p(\beta)$ in $\Delta[\Delta[\mathcal{Y}]]$, following a piecewise smooth trajectory dictated by the IB ODE (16) (which may be considered as a non-autonomous ODE on $\Delta[\Delta[\mathcal{Y}]]$). Due to Conjecture 1 and the IFT, these paths are isolated outside bifurcations. Two (or more) roots may intersect in a continuous bifurcation. If one of the intersecting roots is optimal, then the other must be of a strictly smaller effective cardinality due to the arguments in Section 5.2. If two distinct roots are optimal simultaneously, then $\Delta[\Delta[\mathcal{Y}]]$ contains an entire segment of optimal IB roots, due to Corollary 3. Viewing the IB this way also highlights several subtleties in its calculation. First, parameterizing IB roots with $\Delta[\Delta[\mathcal{Y}]]$ avoids its structural symmetries (Section 5.1). Second, it shows that, a priori, it is possible to follow the path of an optimal root using local techniques (Corollary 4). Third, it highlights that one must compute on enough clusters to detect a bifurcation (Proposition 1). Though obvious in retrospect, this caveat was not given proper attention in the IB literature. Fourth, as we shall now see, cluster-vanishing bifurcations can only be detected by following an optimal root to its collision with a root of smaller effective cardinality. Fifth, this implies (below) that only negative step sizes $\Delta\beta < 0$ should be used to follow an optimal root.

The arguments above imply that cluster-vanishing bifurcations cannot be detected directly by considering kernel directions of the IB operator (5) at the bifurcation, as argued in Section 5.2. Indeed, consider a continuous bifurcation, where roots p_1 and p_2 of respective effective cardinalities $T_1 < T_2$ intersect. These are paths in $\Delta[\Delta[\mathcal{Y}]]$ that coincide at the bifurcation itself, $p_1(\beta_c) = p_2(\beta_c)$, and so in particular are of the same effective cardinality T_1 there. This is in contrast to the situation in Corollary 3, where two *distinct* roots are

simultaneously optimal at β_c , leading to an entire segment of optimal roots. Asking whether a bifurcation is detectable amounts to considering the evaluation of $\ker D(\text{Id} - BA_\beta)$ at a finite-dimensional representation (or “projection”) of p . The Jacobian $D(\text{Id} - BA_\beta)$ of the IB operator (5) is non-singular when evaluated on a T_1 -clustered representation of $p_1(\beta_c)$ in log-decoder coordinates, as noted after Proposition 1. We argue that evaluating it on representations with more clusters $T \gtrsim T_1$ does *not* allow one to detect the bifurcation (even if $T \geq T_2$). See Appendix H for a formal argument. Intuitively, this is because picking a degenerate representation amounts to duplicating clusters of the reduced representation or adding clusters of zero mass (see *reduction* in Section 5.1). Introducing degeneracies to a reduced root adds no information about the problem at hand.

Due to the above, cluster-vanishing bifurcations cannot be detected by following a root p_1 of effective cardinality T_1 through the bifurcation point, but only by following a root p_2 with $T_2 > T_1$ to its collision with p_1 . As discussed after Conjecture 1 (Section 5.1), the Jacobian of $\text{Id} - BA_\beta$ in reduced log-decoder coordinates can then be used to indicate the upcoming collision of p_2 with p_1 , in addition to the root-reduction Algorithm 2. The exact same arguments as above apply also to cluster-merging bifurcations. However, as noted in Section 5.2 (and Appendix F), the stability of a particular IB cluster \hat{x} is a property of the root itself. Thus, these are detectable by standard local techniques at the point of bifurcation. Unlike continuous bifurcations, discontinuous bifurcations are inherently detectable due to the line segment in $\Delta[\Delta[\mathcal{Y}]]$ connecting the roots at the bifurcation (Corollary 3), as long as the IB root is represented on sufficiently many clusters (Proposition 1)—see Figure 9. These results make sense, considering that cluster-vanishing bifurcations are more frequent in practice than other types. Intuitively, branching from a suboptimal root p_1 to an optimal one p_2 is harder than the other way around, just as learning new relevant information is harder than discarding it. Cases where both directions are equally difficult are the exception, as one might expect. This is consistent with the later discussion in Section 6.3 on the stability of optimal IB roots (Appendix G).

When following the path of a reduced IB root (as in Section 4), one would like to ensure that its bifurcations are indeed detectable by BA’s Jacobian. Due to the caveats involved in detecting bifurcations of either type, it is necessary to follow the path as the effective cardinality decreases rather than increases. As a result, we take only negative step sizes $\Delta\beta < 0$, since the effective cardinality of an optimal IB root cannot decrease with β . To see this, first note that the IB curve $I_Y(I_X)$ (1) is concave, and so its slope $1/\beta$ cannot increase with I_X . That is, β cannot decrease with I_X . Second, note that allowing more clusters cannot decrease the X -information $\sum_{\hat{x}} p(\hat{x})H(p(x|\hat{x}))$ achieved by the IB’s optimization variables. Indeed, a T -clustered variable $(p(x|\hat{x}), p(\hat{x}))$ (*not* necessarily a root) can always be considered as $(T + 1)$ -clustered, by adding a cluster of zero mass; cf., the construction of [2] (Section II.A). Thus, the effective cardinality of an optimal root cannot decrease as the constraint I_X on the X -information is relaxed. With both points combined, the effective cardinality cannot decrease with β , as argued. In contrast to the IB, we note that the behavior of RD problems is more complicated since the distortion of each reproduction symbol is fixed a priori; e.g., Example 2.7.3 and Problems 2.8–2.10 in [28].

Returning to discontinuous IB bifurcations, we proceed with the argument of Section 5.2 when continuity fails. That is, consider the reduced form of an optimal IB root, and suppose that either its decoders or its weights (or both) cannot be written as a continuous function of β at β_c . Write $r_{\hat{x}}^+$ and $r_{\hat{x}}^-$ for its distinct decoders as $\beta \rightarrow \beta_c^+$ and $\beta \rightarrow \beta_c^-$, respectively. Similarly, write $p^+(\hat{x})$ and $p^-(\hat{x})$ for its non-zero weights. Consider the tangent RD problem on the reproduction alphabet $\hat{\mathcal{X}} := \{r_{\hat{x}}^+\}_{\hat{x}} \cup \{r_{\hat{x}}^-\}_{\hat{x}} \subset \Delta[\mathcal{Y}]$, as in Section 5.1. See also [4] (Section V), upon which this argument is based. By construction, the IB coincides with its tangent RD problem at the two points $(r_{\hat{x}}^+, p^+(\hat{x}))$, and $(r_{\hat{x}}^-, p^-(\hat{x}))$. Since both points achieve the optimal curve at the same slope value $1/\beta_c$, then the linear segment of distributions connecting these points is also optimal, by Theorem 5. Alternatively, one could apply Corollary 3 directly to the IB problem. Either way, there exists a line segment of optimal IB roots, which pertain to the given slope value. In summary,

Theorem 6. *Let a finite IB problem have a discontinuous bifurcation at $\beta_c \geq 1$. Then, its IB curve (1) has a linear segment of slope $1/\beta_c$.*

Unless the decoder sets $\{r_{\hat{x}}^+\}_{\hat{x}}$ and $\{r_{\hat{x}}^-\}_{\hat{x}}$ are identical, then this is a support-switching bifurcation [6] (Section 6.5), as in Figure 7. *A priori*, the IB roots $(r_{\hat{x}}^+, p^+(\hat{x}))$ and $(r_{\hat{x}}^-, p^-(\hat{x}))$ may achieve the same point in the information plane, in which case the linear curve segment is of length zero. However, we are unaware of such examples. Yet, even if such bifurcations exist, they would be detectable by the Jacobian of BA-IB (when represented on enough clusters), subject to Conjecture 1.

6. First-Order Root Tracking for the Information Bottleneck

Gathering the results of Sections 2–5, we can now not only follow the evolution of an IB root along the first-order Equation (16), but can also identify and handle IB bifurcations. This is summarized by our First-order Root-Tracking algorithm for the IB (IBRT1) in Section 6.1, with some numerical results in Section 6.2. Section 6.3 discusses the basic properties of IBRT1, and mainly the surprising quality of approximations of the IB curve (1) that it produces, as seen in Figure 1. We focus on continuous bifurcations (Section 5.2), since these are far more frequent in our experience than discontinuous ones and are straightforward to handle (see Section 6.3 on the handling of discontinuous bifurcations).

6.1. The IBRT1 Algorithm 5

To assist the reader, we first present a simplified version of IBRT1 as Algorithm 3, with edge cases handled later by Algorithm 4—clarifications follow. When combined, these two form our IBRT1 Algorithm 5, specified below.

We now elaborate on the main steps of the Simplified First-order Root Tracking for the IB (Algorithm 3), following Root Tracking for RD [6] (Algorithm 3). Its purpose is to follow the path of a given IB root $p_{\beta_0}(\hat{x}|x)$ in a finite IB problem. The initial condition $p_{\beta_0}(\hat{x}|x)$ is required to be reduced and IB-optimal. Its optimality is needed below to ensure that the path traced by the algorithm is indeed optimal. The step-size $\Delta\beta$ is negative, for reasons explained in Section 5.3 (Proposition 1 ff.). The cluster mass and cluster merging thresholds are as in the root-reduction Algorithm 2 (Section 5.2).

Denote \tilde{p} (step 3 of Algorithm 3) for the distributions generated from an encoder (see Equation (11) in Section 2). Algorithm 3 iterates over grid points \tilde{p} , with each **while** iteration generating the reduced form of the next grid point, as follows. On step 6, evaluate the IB ODE (16) at the current root \tilde{p} , solving the linear equations numerically. By Conjecture 1 (Section 5.1), the IB ODE has a unique numerical solution v if \tilde{p} is a reduced root and not a bifurcation. Steps 7 and 8 approximate the root at the next grid point at $\beta + \Delta\beta$, by exponentiating Euler method's step (22) (Section 4). Normalization is enforced on step 9, since it is assumed throughout. Off-grid points can be generated by repeating steps 7 through 9 for intermediate $\Delta\beta$ values if desired. The approximate root at $\beta + \Delta\beta$ is reduced on step 11, by invoking the root-reduction Algorithm 2 (Section 5.2). Note that Algorithm 2 returns its input root unmodified unless reducing it numerically. If reduced, then the root is a vector of a lower dimension—either a cluster mass $p(\hat{x})$ has nearly vanished or distinct clusters have nearly merged. To re-gain accuracy, we invoke (on step 14) the Blahut–Arimoto Algorithm 1 for the IB until convergence, on the encoder defined at step 13 by the reduced root. Although BA-IB is invoked near a bifurcation, this does *not* incur a hefty computational cost due to its critical slowing down [4]—see comments at the bottom of Section 5.2. Invoking BA (on step 14) *before* reducing (on step 11) would have inflicted a hefty computational cost to BA-IB due to the nearby bifurcation. Finally, a single BA-IB iteration in decoder coordinates is invoked on the approximate root (step 17), whether reduced earlier or not. This enforces Markovity while improving the order of this method (see Section 4, and Figure 4 in particular). Algorithm 3 continues this way (step 4) until the approximate solution is trivial (single-clustered), or β is non-positive. In the IB, the trivial solution is always optimal for tradeoff values $\beta < 1$. However, here β plays the role of

the ODE’s independent variable instead. Thus, we allow Algorithm 3 to continue beyond $\beta = 1$, as long as $\beta > 0$, which is assumed throughout (the condition $\beta > |\Delta\beta|$ on step 4 ensures that the target β value of the *next* grid point is non-negative). This shall be useful for overshooting—see below.

Algorithm 3 Simplified First-order Root-Tracking for the IB

```

1: function sIBRT1( $p_{Y|X}$ ,  $p_X$ ,  $\beta_0$ ,  $p_{\beta_0}(\hat{x}|x)$ ;  $\Delta\beta$ ,  $\delta_1$ ,  $\delta_2$ )
Input:
    An IB problem definition  $p_{Y|X}$ ,  $p_X$  with  $\forall x p_X(x) > 0$ .
    A reduced IB-optimal root  $p_{\beta_0}(\hat{x}|x)$  at  $\beta_0$ . A step size  $\Delta\beta < 0$ .
    Cluster-mass threshold  $\delta_1$  and cluster-merging threshold  $\delta_2$ , with  $0 < \delta_i < 1$ .
Output: Approximations  $\tilde{p}_{\beta_n}$  of the optimal IB roots  $p_{\beta_n}$  at  $\beta_n := \beta_0 + n\Delta\beta$ .
2:   Initialize  $\beta \leftarrow \beta_0$  and  $results \leftarrow \{\}$ .
3:   Initialize  $\tilde{p} := (\tilde{p}(\hat{x}|x), \tilde{p}(x|\hat{x}), \tilde{p}(y|\hat{x}), \tilde{p}(\hat{x}))$  from  $p_{\beta_0}(\hat{x}|x)$ , via Steps 1.4–1.6.
4:   while  $\beta > |\Delta\beta|$  and  $|\text{supp } \tilde{p}(\hat{x})| > 1$  do      ▷ See main text on stopping condition.
5:     Append  $\tilde{p}$  to  $results$ .
6:      $v := \left( \frac{d \log \tilde{p}(y|\hat{x})}{d\beta}, \frac{d \log \tilde{p}(\hat{x})}{d\beta} \right) \leftarrow$  solve the IB ODE (16) at  $\tilde{p}$ .
7:      $\tilde{p}(y|\hat{x}) \leftarrow \tilde{p}(y|\hat{x}) \exp\left(\Delta\beta \cdot \frac{d \log \tilde{p}(y|\hat{x})}{d\beta}\right)$ 
8:      $\tilde{p}(\hat{x}) \leftarrow \tilde{p}(\hat{x}) \exp\left(\Delta\beta \cdot \frac{d \log \tilde{p}(\hat{x})}{d\beta}\right)$  ▷ Exponentiate the linear approximations (22).
9:      $(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x})) \leftarrow$  normalize  $(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x}))$ 
10:     $old\_dim \leftarrow \dim \tilde{p}(\hat{x})$ 
11:     $(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x})) \leftarrow$  REDUCE ROOT( $\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x}); \delta_1, \delta_2$ ).      ▷ Algorithm 2.
12:    if  $old\_dim \neq \dim \tilde{p}(\hat{x})$  then      ▷ Root was reduced due to bifurcation.
13:       $\tilde{p}(\hat{x}|x) \leftarrow$  the encoder defined by  $(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x}))$ , via Steps 1.7–1.8.
14:       $\tilde{p} \leftarrow$  BA-IB( $\tilde{p}(\hat{x}|x); p_{Y|X}$ ,  $p_X$ ,  $\beta + \Delta\beta$ ).
      ▷ Ensure accuracy of the reduced root, using BA-IB Algorithm 1 till convergence.
15:    end if
16:     $\beta \leftarrow \beta + \Delta\beta$ .
17:     $\tilde{p} \leftarrow$  BA $_{\beta}$ ( $\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x})$ )      ▷ A single BA-IB iteration in decoder coordinates.
18:  end while
19:  Append  $\tilde{p}$  to  $results$ .
20:  return  $results$ .
21: end function

```

With that, there are caveats in Algorithm 3, which stem from passing too far or close to a bifurcation. For one, suppose that the error accumulated from the true solution is too large for a bifurcation to be detected. The approximations generated by the algorithm will then overshoot the bifurcation. Namely, it will proceed with more clusters than needed until the conditions for reduction are met later on (see Section 6.3 below), as demonstrated by the two sparse grids in Figure 10 (Section 6.2). For another, suppose that the current grid point \tilde{p} is too close to a bifurcation. This might happen due to a variety of numerical reasons, e.g., thresholds δ_1, δ_2 too small, or due to the particular grid layout. The coefficients matrix $I - D_{\log p(y|\hat{x}), \log p(\hat{x})} BA_{\beta}$ of the IB ODE (16) (which is the Jacobian of the IB operator (5)) would then be ill-conditioned, typically resulting in very large implicit numerical derivatives v on step 6; cf., Conjecture 1 ff. in Section 5.1. Any inaccuracy in v might then send the next grid point astray, derailing the algorithm from there on (e.g., inaccuracies due to the accumulated approximation error or due to the error caused by computing implicit derivatives in the vicinity of a bifurcation—see Figure 3 (top) in Section 3). Indeed, the derivatives $\frac{dx}{d\beta} = -(D_x F)^{-1} D_{\beta} F$ defined by the implicit ODE (7) are in general unbounded near a bifurcation of F (in our case, $D_x F$ is always non-singular outside bifurcations, due to Conjecture 1 and the use of reduced coordinates). This can be seen in Figure 2 (Section 2) for example, where the derivatives “explode” at

the bifurcation’s vicinity. See also [6] (Section 7.2) on the computational difficulty incurred by a bifurcation. While overshooting a bifurcation is not a significant concern for our purposes (see Section 6.3), passing too close to one is. The latter is important, especially when the step size $|\Delta\beta|$ is small. While decreasing $|\Delta\beta|$ generally improves the error of Euler’s method, it also makes it easier for the approximations to come close to a bifurcation, thus potentially worsening the approximation dramatically if it derails. This motivates one to consider how singularities of the IB ODE (16) should be handled.

Algorithm 4 A heuristic for handling singularities of the IB ODE (16)

1: **function** HANDLE SINGULARITY($p_{Y|X}$ p_X , $(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x}))$, v , β)

Input:

 An IB problem definition $p_{Y|X}$ p_X , with $\forall x$ $p_X(x) > 0$.

 An approximate root $(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x}))$ of the given problem, near a singularity of the IB ODE (16).

 Approximate numerical derivatives $v := \left(\frac{d \log \tilde{p}(y|\hat{x})}{d\beta}, \frac{d \log \tilde{p}(\hat{x})}{d\beta} \right)$ at the given root.

 The $\beta > 0$ value of the next (output) grid point.

Output: An approximate IB root \tilde{p} at β on one fewer cluster.

2: $\hat{x}', \hat{x}'' \leftarrow$ the two indices \hat{x} of largest $\left\| \frac{d \log \tilde{p}(y|\hat{x})}{d\beta} \right\|_{\infty}$ value (norm of y -indexed vectors).

3: $\tilde{p}(y|\hat{x}') \leftarrow \frac{1}{2} \cdot (\tilde{p}(y|\hat{x}') + \tilde{p}(y|\hat{x}''))$ \triangleright Replace fastest-moving clusters by their mean.

4: Erase \hat{x}'' from the decoder $\tilde{p}(y|\hat{x})$.

5: $\tilde{p}(\hat{x}') \leftarrow \tilde{p}(\hat{x}') + \tilde{p}(\hat{x}'')$

6: Erase \hat{x}'' from the marginal $\tilde{p}(\hat{x})$.

7: $\tilde{p}(\hat{x}|x) \leftarrow$ the encoder generated from $(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x}))$, via Steps 1.7–1.8.

\triangleright A new encoder on one cluster *less* than the input.

8: $\tilde{p} \leftarrow$ BA-IB($\tilde{p}(\hat{x}|x); p_{Y|X}$ p_X , β). \triangleright Re-gain accuracy, by the BA-IB Algorithm 1.

9: **return** \tilde{p}

10: **end function**

Next, we elaborate on our heuristic for handling singularities of the IB ODE (16), Algorithm 4. The inputs of this heuristic are defined as in Algorithm 3. It starts with the assumption that the coefficients matrix $I - D_{\log p(y|\hat{x}), \log p(\hat{x})} BA_{\beta}$ of the IB ODE (16) is nearly singular at the current grid point \tilde{p} due to a nearby bifurcation (although a priori the Jacobian $D_{\log p(y|\hat{x}), \log p(\hat{x})} (Id - BA_{\beta})$ may be singular also due to other reasons, by Conjecture 1 it is non-singular at the approximations generated so far since they are assumed to be in their reduced form—see Section 5.1). As a result, the implicit derivatives v at \tilde{p} are not to be used directly to extrapolate the next grid point, as explained above. Instead, we use them to identify the two fastest moving clusters, on step 2 of Algorithm 4 (while this can be refined to handle more than two fast-moving clusters at once, that is not expected to be necessary for typical bifurcations). These are replaced by a single cluster (steps 3 through 6), resulting in an approximate root on one fewer cluster. To re-gain accuracy, the BA-IB Algorithm 1 is then invoked (on step 8) on the encoder generated (on step 7) from the latter root, thereby generating the next grid point. If the fastest-moving clusters have merged (in the true solution) by the next grid point, then the output of Algorithm 4 will be an IB-optimal root if its input grid point is so. Namely, the branch followed by the algorithm remains an optimal one. Otherwise, if these clusters merge shortly after the next grid point, then Algorithm 4 yields a sub-optimal branch. However, optimality is re-gained shortly afterward since the sub-optimal branch collides and merges with the optimal one in continuous IB bifurcations (Section 5.3). Figure 10 below demonstrates Algorithm 4. cf., [6] (Section 3.2) on the similar heuristic in root tracking for RD, which may also lose optimality near a bifurcation and re-gain it shortly after.

Algorithm 5 First-order Root Tracking for the IB (IBRT1)

```

1: function IBRT1( $p_{Y|X}$   $p_X$ ,  $\beta_0$ ,  $p_{\beta_0}(\hat{x}|x)$ ;  $\Delta\beta$ ,  $\delta_1$ ,  $\delta_2$ ,  $\delta_3$ )
Input:
  An IB problem definition  $p_{Y|X}$   $p_X$  with  $\forall x p_X(x) > 0$ .
  A reduced IB-optimal root  $p_{\beta_0}(\hat{x}|x)$  at  $\beta_0$ . A step size  $\Delta\beta < 0$ .
  Thresholds  $0 < \delta_1, \delta_2 < 1$  for the root-reduction Algorithm 2 (cluster mass and merging).
  A threshold  $0 < \delta_3 < 1$  for eigenvalues' singularity.
Output: Approximations  $\tilde{p}_{\beta_n}$  of the optimal IB roots  $p_{\beta_n}$  at  $\beta_n := \beta_0 + n\Delta\beta$ .
2: Initialize  $\beta \leftarrow \beta_0$  and  $results \leftarrow \{\}$ .
3: Initialize  $\tilde{p} := (\tilde{p}(\hat{x}|x), \tilde{p}(x|\hat{x}), \tilde{p}(y|\hat{x}), \tilde{p}(\hat{x}))$  from  $p_{\beta_0}(\hat{x}|x)$ , via Steps 1.4–1.6.
4: while  $\beta > |\Delta\beta|$  and  $|\text{supp } \tilde{p}(\hat{x})| > 1$  do
5:   Append  $\tilde{p}$  to  $results$ .
6:    $v := \left( \frac{d \log \tilde{p}(y|\hat{x})}{d\beta}, \frac{d \log \tilde{p}(\hat{x})}{d\beta} \right) \leftarrow$  solve the IB ODE (16) at  $\tilde{p}$ .
7:    $eigs \leftarrow \text{eig}(I - S)|_{\tilde{p}} \quad \triangleright$  Test ODE for singularity, using  $S$  (17) from Lemma 1.
8:   if  $(\min_{v \in eigs} |v|) < \delta_3$  then  $\triangleright$  ODE is nearly singular.
9:      $\tilde{p} \leftarrow \text{HANDLE SINGULARITY}(p_{Y|X} p_X, (\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x})), v, \beta + \Delta\beta)$ 
 $\triangleright$  Handle an otherwise undetected singularity using Algorithm 4.
10:   else
11:      $\tilde{p}(y|\hat{x}) \leftarrow \tilde{p}(y|\hat{x}) \exp\left(\Delta\beta \cdot \frac{d \log \tilde{p}(y|\hat{x})}{d\beta}\right)$ 
12:      $\tilde{p}(\hat{x}) \leftarrow \tilde{p}(\hat{x}) \exp\left(\Delta\beta \cdot \frac{d \log \tilde{p}(\hat{x})}{d\beta}\right)$ 
13:      $(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x})) \leftarrow \text{normalize}(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x}))$ 
14:      $old\_dim \leftarrow \text{dim } \tilde{p}(\hat{x})$ 
15:      $(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x})) \leftarrow \text{REDUCE ROOT}(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x}); \delta_1, \delta_2)$ .
16:     if  $old\_dim \neq \text{dim } \tilde{p}(\hat{x})$  then
17:        $\tilde{p}(\hat{x}|x) \leftarrow$  encoder defined from  $(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x}))$ , via Steps 1.7–1.8.
18:        $\tilde{p} \leftarrow \text{BA-IB}(\tilde{p}(\hat{x}|x); p_{Y|X} p_X, \beta + \Delta\beta)$ .
19:     end if
20:   end if
21:    $\beta \leftarrow \beta + \Delta\beta$ .
22:    $\tilde{p} \leftarrow \text{BA}_\beta(\tilde{p}(y|\hat{x}), \tilde{p}(\hat{x}))$ 
23: end while
24: Append  $\tilde{p}$  to  $results$ .
25: return  $results$ .
26: end function

```

The heuristic Algorithm 4 is motivated by cluster-merging bifurcations. In these, the implicit derivatives are very large only at the coordinates $\frac{d \log p(y|\hat{x})}{d\beta}$ of the points colliding in $\Delta[\mathcal{Y}]$ (note that cluster masses barely change in the vicinity of a cluster merging, until the point of bifurcation itself). While intended for cluster-merging bifurcations, this heuristic works nicely in practice also for cluster-vanishing ones. To see why, note that one can always add a cluster of zero mass to an IB root without affecting the root's essential properties, regardless of its coordinates in $\Delta[\mathcal{Y}]$ (cf., Section 5.1 on reduction in the IB). Therefore, a numerical algorithm may, in principle, do anything with the coordinates $p(y|\hat{x}) \in \Delta[\mathcal{Y}]$ of a nearly vanished cluster \hat{x} , $p(\hat{x}) \simeq 0$, without affecting the approximation's quality too much. Thus, for numerical purposes, one may treat a cluster-vanishing bifurcation as a cluster-merging one. Conversely, in a cluster-merging bifurcation, a numerical algorithm may, in principle, zero the mass of one cluster while adding it to the remaining cluster, again without affecting the approximation's quality too much. To conclude, for numerical purposes, cluster vanishing is very similar to cluster merging. A variety of treatments between these extremities may be possible by a numerical algorithm. Empirically, we have observed that our ODE-based algorithm treats both as cluster-merging bifurcations. To our

understanding, this is because our algorithm operates in decoder coordinates, unlike the BA-IB Algorithm 1, for example, which operates in encoder coordinates.

Finally, we combine the simplified root-tracking Algorithm 3 with the heuristic Algorithm 4 for handling singularities, yielding our IBRT1 Algorithm 5. It follows the lines of the simplified Algorithm 3, except that after solving for the implicit derivatives on step 6, we test the IB ODE (16) for singularity. To that end, we propose using the matrix S (17) (from Lemma 1 in Section 3), since its order $T \cdot |\mathcal{Y}|$ is smaller than the order $T \cdot (|\mathcal{Y}| + 1)$ of the ODE’s coefficients matrix. This might make it computationally cheaper to test for singularity (on steps 7 and 8 of Algorithm 5). Our heuristic Algorithm 4 is invoked (on step 9) if the ODE (16) is found to be nearly singular, otherwise proceeding as in Algorithm 3.

6.2. Numerical Results for the IBRT1 Algorithm 5

To demonstrate the IBRT1 Algorithm 5, we present the numerical results used to approximate the IB curve in Figure 1 (Section 1)—see Section 6.3 below on the approximation quality and the algorithm’s basic properties. This example was chosen both because it has an analytical solution (Appendix E) and because it allows one to get a good idea of the bifurcation handling added (in Section 6.1) on top of the modified Euler method (from Section 4).

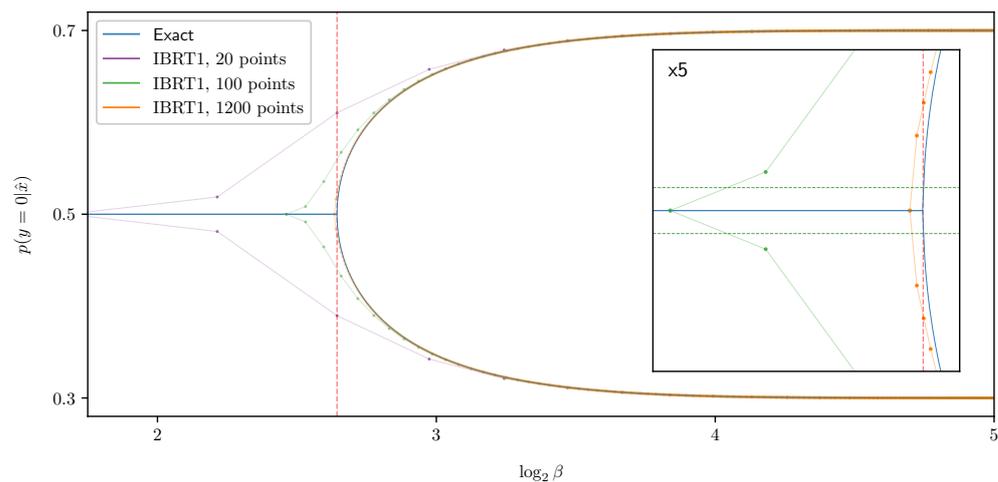


Figure 10. Clusters of the approximate IB roots generated by the IBRT1 Algorithm 5 for several step-sizes, on top of the exact solutions of BSC(0.3) with a uniform source (Appendix E). Carefully note that only a *single* IB root is plotted here; its two clusters merge at β_c (dashed red vertical), as seen in Figure 6 (Section 5.2). At 20 and 100 grid points, the approximations overshoot the bifurcation, terminating due to (approximate) cluster collision, while on 1200 grid points, the approximations pass too close to the bifurcation, terminating due to the nearby singularity. This can be seen in the **inset to the right**: The leftmost green marker has passed the cluster-merging threshold (dashed green lines), and so was numerically reduced to the trivial (single-clustered) solution by the root-reduction Algorithm 2. On the other hand, the orange markers to the right are still far from the cluster-merging threshold; the leftmost one was reduced by the singularity-handling heuristic Algorithm 4 since the IB ODE (16) is nearly singular there. Indeed, the numerical derivative is about five orders of magnitude larger there than at the algorithm’s initial condition (see Figure 2) due to the bifurcation’s proximity. The leftmost green and orange markers were drawn *after* the reductions took place. See main text and Section 6.1 for details, Figure 11 for errors, and Figure 1 (in Section 1) for the approximate IB curves. The marginals $p(\hat{x})$ are not shown, as these barely deviate from their true value in this problem. For each step-size $\Delta\beta$, the algorithm was initialized at the problem’s exact solution at $\beta = 2^5$, with thresholds set to $\delta_i = 10^{-2}$, for $i = 1, 2, 3$. The lines connecting consecutive markers are for visualization only.

We discuss the numerical examples of this Section in light of the explanations provided in the previous Section 6.1. The error of the IBRT1 Algorithm 5 generally improves as the step-size $|\Delta\beta|$ becomes smaller, as expected. The single BA-IB iteration added to Euler’s method (in Section 4) typically allows one to achieve the same error by using much fewer grid points, thus lowering computational costs. For example, the two denser grids in Figure 10 require about an order of magnitude fewer points to achieve the same error compared to Euler’s method for the IB; this can be seen from Figure 4 (Section 4).

In sparse grids, the approximations often pass too far away from a bifurcation for the root-reduction Algorithm 2 to detect it. When overshooting it, the conditions for numerical reduction are generally met later on, as discussed in Section 6.3 below. Decreasing $|\Delta\beta|$ further often leads the approximations too close to a bifurcation, as can be seen in the densest grid of Figure 10. The implicit derivatives are typically very large at the proximity of a bifurcation, while the least accurate there (see Section 6.1). As these might send subsequent grid points off-track, the heuristic Algorithm 4 is invoked to handle the nearby singularity (see inset of Figure 10). As noted earlier, the computational difficulty in tracking IB roots (or root tracking in general) stems from the presence of a bifurcation, manifested here by large approximation errors in its vicinity. While the algorithm’s error peaks at the bifurcation, it typically decreases afterward when overshooting, as seen in Figure 11. The reasons for this are discussed below in Section 6.3.

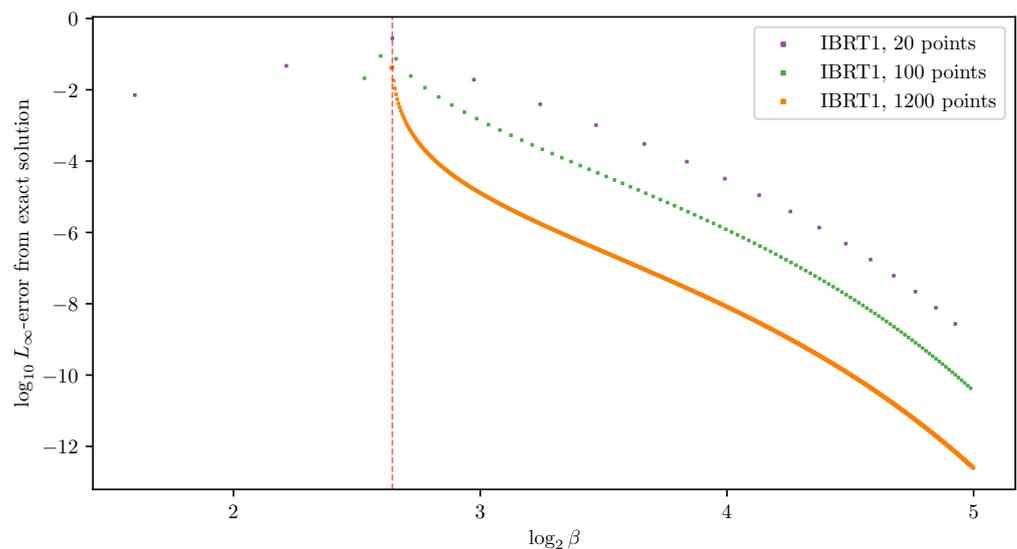


Figure 11. The error of the IBRT1 Algorithm 5 from the exact solution for several step-sizes. The figure shows the (log-) L_∞ -error of the numerical approximations in Figure 10 from the exact solutions; the error is measured as in Figure 3 (bottom). Increasing the grid density decreases the error, as one might expect. While the error peaks at the bifurcation (dashed red vertical), it decreases afterward—see main text and Section 6.3 below. The rightmost marker for each grid density is missing since the initial error is zero.

6.3. Basic Properties of the IBRT1 Algorithm 5 and Why It Works

Apart from presenting the basic properties of the IBRT1 Algorithm 5, the primary purpose of this section is to understand why it approximates the problem’s true IB curve (1) so well, despite its apparent errors in approximating the IB roots. While shown here only in Figures 1 and 10 (in Sections 1 and 6.2), this behavior is consistent in the few numerical examples that we have tested. We offer an explanation why this may be true in general.

To understand why the IBRT1 Algorithm 5 approximates the true IB curve (1) so well, we first explain why overshooting is not a significant concern, as noted earlier in Section 6.1. To that end, consider the implicit ODE (7)

$$\frac{dx}{d\beta} = -(D_x F)^{-1} D_\beta F,$$

from Section 1. As long as $D_x F$ and $D_\beta F$ at its right-hand side are well-defined, it defines a vector field on the entire *phase space* of admissible x values, at least when $D_x F$ is non-singular. That is, even for x 's which are *not* roots (6) of F . Ignoring several technicalities, the IB ODE (16) therefore defines a vector field also *outside* IB roots (although at a reduced root singularities of the IB ODE (16) coincide with IB bifurcations, the IB's vector field might a priori be singular elsewhere). Indeed, due to Conjecture 1, the Jacobian of the IB operator $Id - BA_\beta$ (5) is non-singular in the vicinity of a reduced root ($D_{\log p(y|\hat{x}), \log p(\hat{x})} BA_\beta$ (13) is continuous in the distributions defining it, and thus so are its eigenvalues, under mild assumptions—cf., Lemma A1 in Appendix A). Now, suppose that p_β is an optimal IB root, and consider a point $p' \neq p_\beta$ in its vicinity. An argument based on a strong notion of Lyapunov stability (in Appendix G) shows that p' flows along the IB's vector field towards p_β in regions that do not contain a bifurcation, though only if flowing in *decreasing* β as done by our IBRT Algorithm 5. An approximation p' would then be “pulled” towards the true root. Stability in decreasing (rather than increasing) β values is very reasonable, considering that p_β follows a path of decreasingly informative representations as β decreases. Indeed, all the paths to oblivion lead to one place—the trivial solution, whose representation in reduced coordinates is unique. As a result, a numerical approximation p' would gradually settle in the vicinity of the true root p_β as seen in Figures 10 and 11, so long as p_β does not change much and the step-size $|\Delta\beta|$ is small enough. While this explanation obviously breaks near a bifurcation, it does suggest that the approximation error should decrease when overshooting it (see Section 6.1), once the true reduced root has settled down. In a sense, overshooting is similar to being in the right place but at the wrong time.

The above suggests that the IBRT1 Algorithm 5 should generally approximate the true IB curve (1) well, despite its errors in approximating IB roots. To see this, note that while β^{-1} is the slope of the optimal curve (1) of the IB [1] (Equation (32)), for the IB ODE (16) it is merely an independent “time-like” variable. When solving for the optimal curve (1), one is not interested in an optimal root or in its β value, but rather in its image ($I(X; \hat{X}), I(Y; \hat{X})$) in the information plane. As a result, achieving the optimal roots but with the wrong β values does yield the true IB curve (1), as required. This is the reason that the true curve (1) is achieved in Figure 1 (Section 1) even on sparse grids, despite the apparent approximation errors in Figures 10 and 11 (Section 6.2). With that, we expect the approximate IB curve produced by the IBRT1 Algorithm 5 to be of lesser quality when there are more than two possible labels y . To see why, note that the space $\Delta[\mathcal{Y}]$ traversed by approximate clusters is not one-dimensional when $|\mathcal{Y}| > 2$, and so it is possible to maneuver around the clusters of an optimal root.

Next, we briefly discuss the basic properties of the IBRT1 Algorithm 5. Its **computational complexity** is determined by the complexity of a single grid point. The latter is readily seen to be dominated by the complexity $O(T^2 \cdot |\mathcal{Y}|^2 \cdot (|\mathcal{X}| + T \cdot |\mathcal{Y}|))$ of computing the coefficients matrix of the IB ODE (16) and of solving it numerically (on step 6). To that, one should add the complexity of the BA-IB Algorithm 1 each time a root is reduced. However, the critical slowing down of BA-IB [4] is avoided since we reduce the root before invoking BA-IB (see Section 5.2). The complexity is only linear in $|\mathcal{X}|$ thanks to the choice of decoder coordinates. Had we chosen one of the other coordinate systems in Section 2, then solving the ODE would have been cubic in $|\mathcal{X}|$ rather than linear (see there). The **computational difficulty** in following IB roots stems from the existence of bifurcations (Section 4), as it generally is with following an operator's root [6] (Section 7.2).

As noted in Section 4, **convergence guarantees** can be derived for Euler's method for the IB when away from bifurcation, in terms of the step-size $|\Delta\beta|$, in a manner similar to [6] (Theorem 5) for RD. These imply similar guarantees for the IBRT1 Algorithm 5, since adding a single BA-IB iteration in our modified Euler method improves its order (see there). These details are omitted for brevity, however.

For a numerical method of order $d > 0$ (see Section 4) with a fixed step-size $|\Delta\beta|$ and a fixed computational cost per grid point, the **cost-to-error tradeoff** is given by

$$error \propto cost^{-d}, \quad (26)$$

as in [6] (Equation (3.6)), when $|\Delta\beta|$ is small enough. See [26] for example. Figure 3.4 in [6] demonstrates for RD that methods of higher order achieve a better tradeoff, as expected, as in the fixed-order Taylor methods they employ. Since computing implicit derivatives of higher orders requires the calculation of many more derivative tensors of $Id - BA_\beta$ (5) than done here [6] (Section 2.2), we have used only first-order derivatives for simplicity. However, while the vanilla Euler method for the IB is of order $d = 1$, the discussion in Section 4 (and Figure 4 in particular) suggests that the order d of the modified Euler method used by the IBRT1 Algorithm 5 is nearly twice that; cf., Section 6.2 and Appendix D.

With that, we comment on the behavior of the IBRT1 Algorithm 5 at **discontinuous bifurcations**. Consider the problem in Figure 7 (Section 5.3), for example. When Algorithm 5 follows the optimal 2-clustered root there, the Jacobian's singularity (in Figure 9) is detectable by it because the step size $\Delta\beta$ is negative (see the discussion in Section 5.3 there). Indeed, due to Conjecture 1 ff., the algorithm can detect discontinuous bifurcations in general. Whether a particular discontinuous bifurcation is detected by Algorithm 5 in practice depends on the details, of course, as with continuous bifurcations (e.g., on the threshold value δ_3 for detecting singularity and on the precise grid point layout). Indeed, the details may or may not cause a particular example to be detected by the conditions on steps 7 and 8 (in Algorithm 5). If missed, Algorithm 5 will continue to follow the 2-clustered root in Figure 7 to the left of the bifurcation, where it is sub-optimal, just as BA-IB with reverse deterministic annealing would. Once detected, though, one may wonder whether the heuristic Algorithm 4 works well also for discontinuous bifurcations. The example of Figure 7 has just one single-clustered root to the left of the bifurcation. Thus, the BA-IB Algorithm 1 invoked on step 8 (of Algorithm 4) must converge to it. However, there may generally be more than a single root of smaller effective cardinality to the left of the bifurcation, to which BA-IB may converge. The handling of discontinuous bifurcations is left to future work. Such handling is expected to be easier in the IB than in RD, since, in contrast to RD, the effective cardinality of an optimal IB root cannot decrease with β (bottom of Section 5.3). See Problems 2.8–2.10 in [28] for counter-examples in RD. This makes detecting discontinuous bifurcations easier in the IB and is also expected to assist with their handling.

We list the **assumptions** used along the way for reference. These are needed to guarantee the optimality of the IBRT1 Algorithm 5 at the limit of small step-sizes $|\Delta\beta|$, except at a bifurcation's vicinity. In Section 1, it was assumed without loss of generality that the input distribution p_X is of full support, $p(x) > 0$ for every x (otherwise, one may remove symbols x with $p_X(x) = 0$ from the source alphabet). The requirement $p(y|x) > 0$ was added in Section 3 as a sufficient technical condition for exchanging to logarithmic coordinates (Lemma A1 in Appendix A), and could perhaps be alleviated in alternative derivations. Together, these are equivalent to having a never-vanishing IB problem definition, $p(y|x)p(x) > 0$ for every x and y . The algorithm's initial condition is assumed to be a reduced and optimal IB root, since reduction is needed by Conjecture 1 in Section 5.1. Finally, the given IB problem is assumed to have only continuous bifurcations, except perhaps for its first (leftmost) one. While these assumptions are sufficient to guarantee optimality, we note that milder conditions might do in a particular problem.

7. Concluding Remarks

The IB is intimately related to several problems in adjacent fields [3], including coding problems, inference, and representation learning. Despite its importance, there are surprisingly few techniques to solve it numerically. This work attempts to fill this gap by exploiting the dynamics of IB roots.

The end result of this work is a new numerical algorithm for the IB, which follows the path of a root along the IB's optimal tradeoff curve (1). A combination of several novelties was required to achieve this goal. First, the dynamics underlying the IB curve (1) obeys an ODE [10]. Following the discussion around Conjecture 1 (in Section 5.1), the existence of such a dynamics stems from the analyticity of the IB's fixed-point Equations (2)–(4), thus

typically resulting in piece-wise smooth dynamics of IB roots. Several natural choices of a coordinate system for the IB were considered, both for computational purposes and to facilitate a clean treatment of IB bifurcations below. The IB's ODE (16) was derived anew in appropriate coordinates, allowing an efficient computation of implicit derivatives at an IB root. Combining BA-IB with Euler's method yields a modified numerical method whose order is higher than either.

Second, one needs to understand where the IB ODE (16) is *not* obeyed, thereby violating the differentiability of an optimal root with respect to β . To that end, one not only needs to detect IB bifurcations but also needs to identify their type in order to handle them properly. Unlike standard techniques, our approach is to remove redundant coordinates, following root tracking for RD [6] (see Section 1). To achieve a reduction, we follow the arguably better definition of the IB in [20]. Namely, a finite IB problem is an RD problem on the continuous reproduction alphabet $\Delta[\mathcal{Y}]$. Therefore, the IB may be intuitively considered as a method of lossy compression of the information on Y embedded in X . Viewing a finite IB problem as an infinite RD problem suggests a particular choice of a coordinate system for the IB, which enables reduction in the IB; this extends reduction in RD [6]. Furthermore, this point of view highlights subtleties of finite dimensionality in computing representations of IB roots. To our understanding, these subtleties hindered the understanding of IB bifurcations throughout the years.

Combining the above allows us to translate an understanding of IB bifurcations to a new numerical algorithm for the IB (the IBRT1 Algorithm 5). There are several directions that one could consider to improve our algorithm. Near bifurcations, one could improve its handling of discontinuous bifurcations. While we used implicit derivatives only of the first order for simplicity, higher-order derivatives generally offer a better cost-to-error tradeoff when away from bifurcations. See also [6] (Section 3.4) on possible improvements for following an operator's root.

Funding: This work was partially funded by the Israel Science Foundation grant 1641/21.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The author is grateful to Or Ordentlich for helpful conversations and for his support, and to Noam and Dafna Agmon for their unwavering support throughout this journey. The author thanks the late Naftali Tishby for insightful conversations and Etam Benger for his involvement during the early stages of this work. The author thanks the reviewers for their helpful comments.

Conflicts of Interest: The author declares no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

IB	Information Bottleneck
RD	Rate Distortion
IFT	Implicit Function Theorem
ODE	Ordinary Differential Equation
BA	Blahut–Arimoto

Appendix A. The BA-IB Operator in Decoder Coordinates

For reference, we give an explicit expression for the BA-IB operator in decoder coordinates, defined in Section 2.

Denote by $p_{Y|\hat{X}}$ and $p_{\hat{X}}$ the vectors whose coordinates are $(p(y|\hat{x}), p(\hat{x}))$. We denote the evaluation of BA_{β} at this point by $BA_{\beta}[p_{Y|\hat{X}}, p_{\hat{X}}]$. Its output is again a decoder–marginal

pair, whose coordinates are denoted, respectively, $BA_\beta[\mathbf{p}_{Y|\hat{X}}, \mathbf{p}_{\hat{X}}](y|\hat{x})$ and $BA_\beta[\mathbf{p}_{Y|\hat{X}}, \mathbf{p}_{\hat{X}}](\hat{x})$. Explicitly, BA_β in decoder coordinates is given by

$$\begin{aligned}
 BA_\beta[\mathbf{p}_{Y|\hat{X}}, \mathbf{p}_{\hat{X}}](y|\hat{x}) &:= \sum_x \frac{p(y|x)p(x)}{Z(x, \beta)} \exp \{ -\beta D_{KL}[p(y'|x)||p(y'|\hat{x})] \} \quad \text{and} \\
 BA_\beta[\mathbf{p}_{Y|\hat{X}}, \mathbf{p}_{\hat{X}}](\hat{x}) &:= \sum_x \frac{p(\hat{x})p(x)}{Z(x, \beta)} \exp \{ -\beta D_{KL}[p(y'|x)||p(y'|\hat{x})] \},
 \end{aligned}
 \tag{A1}$$

where $Z(x, \beta)$ is defined in terms of $p(y|\hat{x})$ and $p(\hat{x})$ as in the IB's encoder Equation (2) (Section 1).

The following lemma is handy when exchanging to logarithmic coordinates in Section 3.

Lemma A1. *Let $p(y|x)p(x)$ define a finite IB problem, such that $p(y|x) > 0$ for every x and y . Let $p(y|\hat{x})$ be the decoder of an IB root, and \hat{x}' such that $p(\hat{x}') > 0$. Then $p(y|\hat{x}') > 0$ for every y .*

Proof of Lemma A1. This follows immediately from the IB's decoder Equation (3), since $p(x|\hat{x}')$ is a well-defined normalized conditional probability distribution if $p(\hat{x}') > 0$. □

Appendix B. The First-Order Derivative Tensors of Blahut–Arimoto for the IB

We calculate the first-order derivative tensors of the Blahut–Arimoto operator BA_β in log-decoder coordinates (see Sections 2 and 3). Namely, its Jacobian matrix $D_{\log p(\hat{x}|x), \log p(\hat{x})} BA_\beta$, and the vector $D_\beta BA_\beta$ of its partial derivatives with respect to β . See also Appendix A for explicit formulae of BA_β in decoder coordinates.

While these are “just” differentiations, many subtleties are involved in getting the math right. For example, one needs to correctly identify the inputs and outputs of BA_β , when considered as an operator on log-decoder coordinates. For another, one must take special care as to which variable depends on which, and especially on which they do *not* depend, since multiple variables are involved. Above all, these calculations require a deep understanding of the chain rule. With that, a common caveat in such calculations is that the BA_β operator (and the equations defining it) should be differentiated *before* they are evaluated. While this is obvious for real functions, where $f'(3)$ stands for the derivative function of $f(x)$ evaluated at $x = 3$, for the BA_β operator, this might be obfuscated by the myriad of variables and variable dependencies that comprise it. Although calculating the derivative of BA_β (at an arbitrary point) first and only then evaluating at a fixed point might appear as a mere technical necessity, it is required by this work, for example, when considering the vector field defined by the IB operator (5) in Section 6.3. cf., [6] (Section 5), for the derivative tensors of Blahut’s algorithm [8] for RD, of arbitrary order.

The subtleties involved in these differentiations are discussed in Appendix B.1, with the bulk of the calculations carried out in Appendix B.1.1. The latter are gathered and simplified in Appendix B.2 to obtain the Jacobian matrix $D_{\log p(\hat{x}|x), \log p(\hat{x})} BA_\beta$, and in Appendix B.3 to obtain the partial-derivatives vector $D_\beta BA_\beta$. The results provided here naturally depend on the choice of coordinate system. To compare results between log-decoder and log-encoder coordinates in Section 2 (e.g., in Figure 2), we derive in Appendix B.4 the coordinate-exchange Jacobians between these coordinate systems.

Appendix B.1. Calculation Setups and Partial Derivatives of Unnamed Functions

We explain the mathematical subtleties relevant to the sequel.

As we are interested in the derivatives of the Blahut–Arimoto Algorithm 1 for the IB (in Section 1), we shall follow its notation. Namely, distributions are subscripted i or $i + 1$ by the algorithm’s iteration number. A subscript i is usually considered an input distribution, and a subscript $i + 1$ is usually considered an output distribution, e.g., $p_i(\hat{x})$ or $p_{i+1}(y|\hat{x})$. These need *not* be IB roots but rather are arbitrary distributions. On the other hand, a subscript β denotes a distribution of an IB root at a tradeoff value β , as in $p_\beta(y|\hat{x})$

for a root’s decoders. To avoid subtleties due to zero-mass clusters, we usually assume in the sequel that $p_i(\hat{x}) \neq 0$ for every \hat{x} . cf., Sections 2 and 5.1 on *reduction* in the IB.

It is important to distinguish which variables are dependent and which are independent in a particular calculation, e.g., in Appendix B.3. Since this task is easier for a single real variable (as opposed to distributions, for example), we consider simplifications to the real case. Note that each of the Steps 1.4 through 1.8 defining the BA-IB Algorithm 1 yields a new distribution in terms of already-specified ones. These define *unnamed functions*, whose variables and values are probability distributions. For example, one could have formally defined $p_i(x|\hat{x})$ in 1.5 by the function

$$\mathcal{F}[p_i(\hat{x}|x), p_i(\hat{x})](x, \hat{x}) := p_i(\hat{x}|x)p(x)/p_i(\hat{x}), \tag{A2}$$

where $p_i(\hat{x}|x)$ and $p_i(\hat{x})$ are the variables of \mathcal{F} , and its output is a conditional probability distribution, with x conditioned upon \hat{x} . As the input and representation alphabets \mathcal{X} and $\hat{\mathcal{X}}$ are finite, $N := |\mathcal{X}|$ and $T := |\hat{\mathcal{X}}|$, the arguments $p_i(\hat{x}|x), p_i(\hat{x})$ and values $p_i(x|\hat{x})$ of \mathcal{F} (A2) are merely real vectors. Thus, enumerating the variables x_1, \dots, x_N and $\hat{x}_1, \dots, \hat{x}_T$ allows us to spell out (A2) by its coordinates,

$$\mathcal{F}[p_i(\hat{x}_1|x_1), p_i(\hat{x}_1|x_2), \dots, p_i(\hat{x}_1|x_N), \dots, p_i(\hat{x}_T|x_N), p_i(\hat{x}_1), \dots, p_i(\hat{x}_T)](x, \hat{x}) := p_i(\hat{x}|x)p(x)/p_i(\hat{x}). \tag{A3}$$

While (A3) is too cumbersome to work with, it does highlight that \mathcal{F} is merely a vector of $N \cdot T$ real vector-valued functions, in $T + N \cdot T$ real variables. This allows us to use partial derivatives rather than their infinite-dimensional counterparts (namely, variational derivatives), as in

$$\frac{\partial \mathcal{F}[p_i(\hat{x}|x), p_i(\hat{x})]}{\partial p_i(\hat{x}_j|x_k)} := \lim_{h \rightarrow 0} \frac{\mathcal{F}[p_i(\hat{x}_1|x_1), \dots, p_i(\hat{x}_j|x_k) + h, \dots, p_i(\hat{x}_T)] - \mathcal{F}[\dots, p_i(\hat{x}_j|x_k), \dots]}{h}. \tag{A4}$$

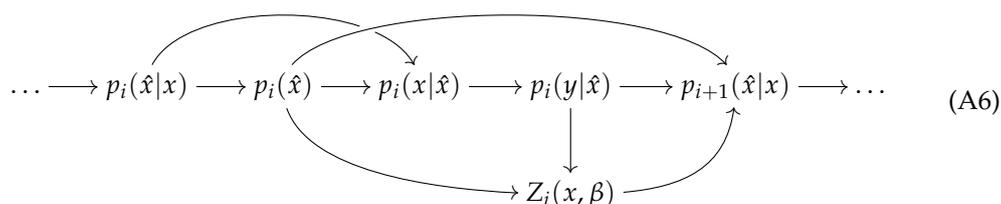
This is the derivative of \mathcal{F} (A3) with respect to a particular (j, k) -entry of its argument, by definition. However, to maintain a concise notation, we shall carry on with un-named function definitions, writing $\partial p_i(x|\hat{x})/\partial p_i(\hat{x}_j|x_k)$ for the partial derivative of (A2) rather than its explicit form (A4). If disoriented, the reader is encouraged to return to the definitions (A4).

We often exchange variables implicitly to logarithmic coordinates, as in Section 3. For example, $\frac{\partial \mathcal{F}[p_i(\hat{x}|x), p_i(\hat{x})]}{\partial \log p_i(\hat{x}_i|x_j)}$ is to be understood as exchanging variables to $u_i(\hat{x}, x) := \log p_i(\hat{x}|x)$, with $\mathcal{G}[u_i(\hat{x}, x), u_i(\hat{x})] := \mathcal{F}[\exp u_i(\hat{x}, x), \exp u_i(\hat{x})]$ now differentiated with respect to its variables $u_i(\hat{x}, x)$ and $u_i(\hat{x})$,

$$\frac{\partial \mathcal{F}[p_i(\hat{x}|x), p_i(\hat{x})]}{\partial \log p_i(\hat{x}_i|x_j)} = \frac{\partial \mathcal{F}[\exp u_i(\hat{x}, x), \exp u_i(\hat{x})]}{\partial u_i(\hat{x}, x)} =: \frac{\partial \mathcal{G}[u_i(\hat{x}, x), u_i(\hat{x})]}{\partial u_i(\hat{x}, x)} \tag{A5}$$

The output of \mathcal{F} may similarly be exchanged to logarithmic coordinates, as in $\log \mathcal{F}[\exp u_i(\hat{x}, x), \exp u_i(\hat{x})]$.

To proceed, carefully note the dependencies between the various variables in a BA-IB iteration, in Steps 1.4 through 1.8. These are summarized compactly by the following diagram:



by their order of appearance in the BA-IB Algorithm 1. This diagram proceeds to both sides by the iteration number i . Each node in (A6) serves both as a function of the nodes preceding it and as a variable for those succeeding it, and so it is a “function-variable”.

To differentiate along the dependencies graph (A6), we shall need the multivariate chain rule

$$\frac{df}{dy} = \frac{\partial f}{\partial y} + \frac{\partial f}{\partial z} \frac{dz}{dy}, \tag{A7}$$

for a function $f(y, z(y))$. As the dependencies graph (A6) involves multiple function variables, such as $z(y)$, we pause on the definition's subtleties. The partial derivative of a function g in several variables x_1, \dots, x_N with respect to its i -th entry is defined by

$$\frac{\partial g}{\partial x_i} := \lim_{h \rightarrow 0} \frac{g(x_1, \dots, x_i + h, \dots, x_N) - g(x_1, \dots, x_i, \dots, x_N)}{h}. \tag{A8}$$

We emphasize that variables $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N$ other than x_i are fixed when calculating $\frac{\partial g}{\partial x_i}$. And so, it makes no difference in (A8) whether or not they depend on x_i , as in $x_j = x_j(x_i)$ for $j \neq i$.

Next, suppose we would like to calculate how changing an input distribution affects some output distribution. This is relevant in Appendix B.2 for example, when considering how a change in a coordinate of an input decoder $p_i(y|\hat{x})$ or marginal $p_i(\hat{x})$ affects a particular coordinate of the output decoder or marginal. For exposition's simplicity, though, suppose that we would like to calculate how a change in the (k_1, k_2) coordinate $p_i(\hat{x}_{k_1}|x_{k_2})$ of an input encoder affects the (j_1, j_2) coordinate $p_{i+1}(\hat{x}_{j_1}|x_{j_2})$ of the output encoder. That is, deriving the rightmost node in (A6) with respect to a coordinate of the leftmost one,

$$\frac{d \log p_{i+1}(\hat{x}_{j_1}|x_{j_2})}{d \log p_i(\hat{x}_{k_1}|x_{k_2})}, \tag{A9}$$

where we have exchanged to logarithmic coordinates to simplify calculations. To calculate (A9), one needs to apply the multivariate chain rule (A7) along all the possible dependencies of the output $\log p_{i+1}(\hat{x}_{j_1}|x_{j_2})$ on the input coordinate $\log p_i(\hat{x}_{k_1}|x_{k_2})$. This amounts to following all the paths in (A6) connecting these two nodes, summing the contributions of every possible path. For example, traversing from the input $p_i(\hat{x}_{k_1}|x_{k_2})$ rightwards at (A6) to $p_i(\hat{x})$, then downwards to $Z_i(x, \beta)$ and then to the output $p_{i+1}(\hat{x}_{j_1}|x_{j_2})$ yields the term

$$\frac{\partial \log p_i(\hat{x}'')}{\partial \log p_i(\hat{x}_{k_1}|x_{k_2})} \frac{\partial \log Z_i(x, \beta)}{\partial \log p_i(\hat{x}'')} \frac{\partial \log p_{i+1}(\hat{x}_{j_1}|x_{j_2})}{\partial \log Z_i(x, \beta)}$$

corresponding to this path, at particular x and \hat{x}'' coordinates. To collect the contribution from every intermediate function variable coordinate, we need to sum the latter over x and \hat{x}'' . Writing down all such paths, one has for (A9),

$$\begin{aligned} & \frac{d \log p_{i+1}(\hat{x}_{j_1}|x_{j_2})}{d \log p_i(\hat{x}_{k_1}|x_{k_2})} \\ &= \frac{\partial \log p_i(\hat{x}'')}{\partial \log p_i(\hat{x}_{k_1}|x_{k_2})} \cdot \left\{ \frac{\partial \log p_{i+1}(\hat{x}_{j_1}|x_{j_2})}{\partial \log Z_i(x, \beta)} \cdot \frac{\partial \log Z_i(x, \beta)}{\partial \log p_i(\hat{x}'')} + \frac{\partial \log p_{i+1}(\hat{x}_{j_1}|x_{j_2})}{\partial \log p_i(\hat{x}'')} \right. \\ &+ \left. \left[\frac{\partial \log p_{i+1}(\hat{x}_{j_1}|x_{j_2})}{\partial \log Z_i(x, \beta)} \cdot \frac{\partial \log Z_i(x, \beta)}{\partial \log p_i(y|\hat{x})} + \frac{\partial \log p_{i+1}(\hat{x}_{j_1}|x_{j_2})}{\partial \log p_i(y|\hat{x})} \right] \cdot \frac{\partial \log p_i(y|\hat{x})}{\partial \log p_i(x'|x')} \cdot \frac{\partial \log p_i(x'|x')}{\partial \log p_i(\hat{x}'')} \right\} \\ &+ \left[\frac{\partial \log p_{i+1}(\hat{x}_{j_1}|x_{j_2})}{\partial \log Z_i(x, \beta)} \cdot \frac{\partial \log Z_i(x, \beta)}{\partial \log p_i(y|\hat{x})} + \frac{\partial \log p_{i+1}(\hat{x}_{j_1}|x_{j_2})}{\partial \log p_i(y|\hat{x})} \right] \cdot \frac{\partial \log p_i(y|\hat{x})}{\partial \log p_i(x'|x')} \cdot \frac{\partial \log p_i(x'|x')}{\partial \log p_i(\hat{x}_{k_1}|x_{k_2})} \end{aligned} \tag{A10}$$

Repeated unbounded variables are understood to be summed over, as in Einstein's summation convention.

Appendix B.1.1. Differentiating along the Dependencies Graph

Next, we differentiate each edge in (the logarithm of) the dependency graph (A6). These are necessary to evaluate derivatives along dependency paths, which underlie the subsequent sections' calculations.

Step 1.4 in the BA-IB Algorithm 1 defines the cluster marginal in terms of the direct encoder,

$$\begin{aligned} \frac{\partial \log p_i(\hat{x})}{\partial \log p_i(\hat{x}'|x')} &\stackrel{\text{Step 1.4}}{=} \frac{1}{p_i(\hat{x})} \sum_x p(x) \frac{\partial}{\partial \log p_i(\hat{x}'|x')} p_i(\hat{x}|x) \\ &= \frac{1}{p_i(\hat{x})} \sum_x p(x) p_i(\hat{x}|x) \frac{\partial \log p_i(\hat{x}|x)}{\partial \log p_i(\hat{x}'|x')} \stackrel{\text{Step 1.5}}{=} p_i(x'|\hat{x}) \cdot \delta_{\hat{x},x'} \end{aligned} \quad (\text{A11})$$

In the first and second equalities we have used the identity $\frac{\partial}{\partial x} y = y \frac{\partial}{\partial x} \log y$ for the differentiation of a function's logarithm, when y is a function of x .

Following the comments around the definition (A8) of a partial derivative, note that Step 1.5 defines the inverse encoder $\log p_i(x|\hat{x})$ as a function of the variables $\log p_i(\hat{x}|x)$ and $\log p_i(\hat{x})$ (and $p(x)$, which we ignore under differentiation). Thus, differentiating this equation with respect to an entry of the variable $\log p_i(\hat{x}|x)$ implies that the entries of the other variable $\log p_i(\hat{x})$ are held fixed, and vice versa. So, for the Bayes rule Step 1.5 we have

$$\frac{\partial \log p_i(x_{j_1}|\hat{x}_{j_2})}{\partial \log p_i(\hat{x})} = \frac{\partial}{\partial \log p_i(\hat{x})} [\log p_i(\hat{x}_{j_2}|x_{j_1}) - \log p_i(\hat{x}_{j_2})] = -\delta_{\hat{x},\hat{x}_{j_2}} \quad (\text{A12})$$

where $\log p_i(\hat{x}_{j_2}|x_{j_1})$ at the right-hand side is different from the variable $\log p_i(\hat{x})$ of differentiation, and so its partial derivative vanishes. Next, differentiating Step 1.5 with respect to a coordinate of its other variable $\log p_i(\hat{x}|x)$,

$$\frac{\partial \log p_i(x_{j_1}|\hat{x}_{j_2})}{\partial \log p_i(\hat{x}'|x')} = \frac{\partial \log p_i(\hat{x}_{j_2}|x_{j_1})}{\partial \log p_i(\hat{x}'|x')} - \frac{\partial \log p_i(\hat{x}_{j_2})}{\partial \log p_i(\hat{x}'|x')} = \delta_{x_{j_1},x'} \cdot \delta_{\hat{x}_{j_2},\hat{x}'} \quad (\text{A13})$$

Using again the logarithmic derivative identity $\frac{\partial}{\partial x} y = y \frac{\partial}{\partial x} \log y$, by the decoder Step 1.6 we have

$$\begin{aligned} \frac{\partial \log p_i(y|\hat{x}'')}{\partial \log p_i(x_{k_1}|\hat{x}_{k_2})} &= \frac{1}{p_i(y|\hat{x}'')} \sum_{x'''} p(y|x''') \frac{\partial}{\partial \log p_i(x_{k_1}|\hat{x}_{k_2})} p_i(x'''|\hat{x}'') \\ &= \frac{1}{p_i(y|\hat{x}'')} \sum_{x'''} p(y|x''') p_i(x'''|\hat{x}'') \delta_{\hat{x}_{k_2},\hat{x}''} \cdot \delta_{x_{k_1},x'''} = \delta_{\hat{x}_{k_2},\hat{x}''} \cdot \frac{p(y|x_{k_1}) p_i(x_{k_1}|\hat{x}'')}{p_i(y|\hat{x}'')} \end{aligned} \quad (\text{A14})$$

Next, consider the KL-divergence term in the Definition 1.7 of the partition function Z_i ,

$$\frac{\partial}{\partial \log p_i(y|\hat{x}'')} D_{KL}[p(y|x'')||p_i(y|\hat{x})] = - \sum_{y'} p(y'|x'') \underbrace{\frac{\partial}{\partial \log p_i(y|\hat{x}'')} \log p_i(y'|\hat{x})}_{\delta_{\hat{x},\hat{x}''} \cdot \delta_{y,y'}} = -\delta_{\hat{x},\hat{x}''} \cdot p(y|x'') \quad (\text{A15})$$

Since the partition Function 1.7 depends on the decoder $p_i(y|\hat{x})$ only via the KL-divergence,

$$\begin{aligned}
 \frac{\partial Z_i(x'', \beta)}{\partial \log p_i(y|\hat{x}'')} &= \frac{\partial}{\partial \log p_i(y|\hat{x}'')} \sum_{\hat{x}} p_i(\hat{x}) \exp \{ -\beta D_{KL}[p(y|x'')||p_i(y|\hat{x})] \} \\
 &= -\beta \sum_{\hat{x}} p_i(\hat{x}) \exp \{ -\beta D_{KL}[p(y|x'')||p_i(y|\hat{x})] \} \frac{\partial}{\partial \log p_i(y|\hat{x}'')} D_{KL}[p(y|x'')||p_i(y|\hat{x})] \\
 &\stackrel{(A15)}{=} \beta p_i(\hat{x}'') \exp \{ -\beta D_{KL}[p(y|x'')||p_i(y|\hat{x}'')] \} p(y|x'') \\
 &\stackrel{\text{Step 1.8}}{=} \beta p_{i+1}(\hat{x}''|x'') Z_i(x'', \beta) p(y|x'')
 \end{aligned} \tag{A16}$$

Hence,

$$\frac{\partial \log Z_i(x'', \beta)}{\partial \log p_i(y|\hat{x}'')} = \beta p_{i+1}(\hat{x}''|x'') p(y|x'') \tag{A17}$$

For the derivative of the partition function with respect to the marginal $p_i(\hat{x})$,

$$\begin{aligned}
 \frac{\partial Z_i(x, \beta)}{\partial \log p_i(\hat{x}')} &\stackrel{\text{Step 1.7}}{=} \frac{\partial}{\partial \log p_i(\hat{x}')} \sum_{\hat{x}} p_i(\hat{x}) \exp \{ -\beta D_{KL}[p(y|x)||p_i(y|\hat{x})] \} \\
 &= \sum_{\hat{x}} p_i(\hat{x}) \exp \{ -\beta D_{KL}[p(y|x)||p_i(y|\hat{x})] \} \frac{\partial \log p_i(\hat{x})}{\partial \log p_i(\hat{x}')} \\
 &= \sum_{\hat{x}} p_i(\hat{x}) \exp \{ -\beta D_{KL}[p(y|x)||p_i(y|\hat{x})] \} \cdot \delta_{\hat{x}, \hat{x}'} \\
 &\stackrel{\text{Step 1.8}}{=} Z_i(x, \beta) \cdot p_{i+1}(\hat{x}'|x)
 \end{aligned} \tag{A18}$$

where the second equality follows from the logarithmic derivative identity. Hence,

$$\frac{\partial \log Z_i(x, \beta)}{\partial \log p_i(\hat{x}')} = p_{i+1}(\hat{x}'|x) \tag{A19}$$

Finally, for the encoder Step 1.8,

$$\log p_{i+1}(\hat{x}'|x') := \log p_i(\hat{x}') - \log Z_i(x', \beta) - \beta D_{KL}[p(y|x')||p_i(y|\hat{x}')] \tag{A20}$$

The first two terms to the right, $p_i(\hat{x})$ and $Z_i(x, \beta)$, take the role of a variable in Step 1.8. In contrast, we consider the last divergence term as a shorthand for summing over $p_i(y|\hat{x})$. Thus, the latter is a variable of (A20). With (A15), we thus have

$$\frac{\partial \log p_{i+1}(\hat{x}'|x')}{\partial \log p_i(y|\hat{x}'')} = \beta \delta_{\hat{x}', \hat{x}''} \cdot p(y|x'). \tag{A21}$$

For the other derivatives of the encoder Step 1.8,

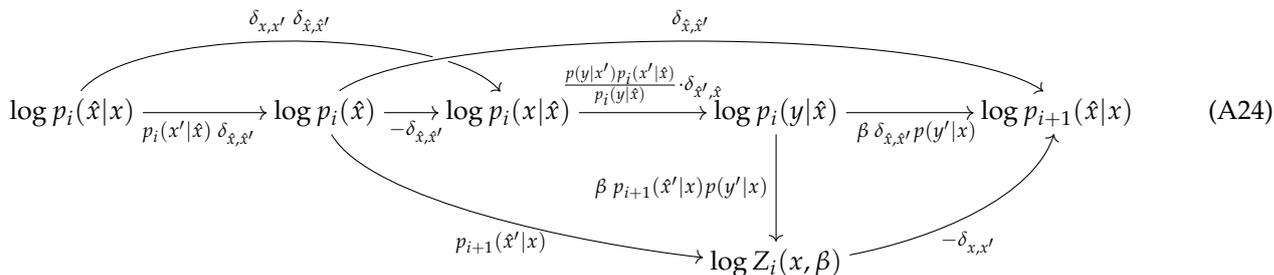
$$\frac{\partial \log p_{i+1}(\hat{x}'|x')}{\partial \log Z_i(x'', \beta)} = -\frac{\partial \log Z_i(x', \beta)}{\partial \log Z_i(x'', \beta)} = -\delta_{x', x''} \tag{A22}$$

and

$$\frac{\partial \log p_{i+1}(\hat{x}'|x)}{\partial \log p_i(\hat{x}')} = \frac{\partial \log p_i(\hat{x})}{\partial \log p_i(\hat{x}')} - \frac{\partial \log Z_i(x, \beta)}{\partial \log p_i(\hat{x}')} - \beta \frac{\partial D_{KL}[p(y|x)||p_i(y|\hat{x})]}{\partial \log p_i(\hat{x}')} = \delta_{\hat{x}, \hat{x}'} \tag{A23}$$

where the variable $p_i(\hat{x})$ of Step 1.8 differs from the variables Z_i and $p_i(y|\hat{x})$, on which the crossed-out terms depend.

We summarize the calculations of this subsection in the following diagram:



A differentiation variable is denoted with commas, at an arrow’s source in this diagram. A coordinate of the function which we differentiate is written without commas, at an arrow’s end, e.g.,

$$\log p_i(\hat{x}|x) \xrightarrow{\frac{\partial \log p_i(x|\hat{x})}{\partial \log p_i(\hat{x}'|x')}} \log p_i(x|\hat{x}) = \dots$$

Appendix B.2. The Jacobian Matrix of BA-IB in Log-Decoder Coordinates

By gathering the results of Appendix B.1.1 and following the lines of Appendix B.1, we calculate the Jacobian matrix (13) (in Section 3) of the Blahut–Arimoto operator BA_β in log-decoder coordinates, defined in Section 2.

The derivative of BA_β in decoder coordinates boils down to the four quantities: the effect $\frac{d \log p_{i+1}(y|\hat{x})}{d \log p_i(y'|\hat{x}'})$ that varying a coordinate $\log p_i(y'|\hat{x}')$ of an input cluster has on a coordinate $\log p_{i+1}(y|\hat{x})$ of an output cluster, the effect $\frac{d \log p_{i+1}(y|\hat{x})}{d \log p_i(\hat{x})}$ that varying an input marginal coordinate $\log p_i(\hat{x})$ has on a coordinate $\log p_{i+1}(y|\hat{x})$ of an output cluster, and so forth. And so, the Jacobian $D_{\log p(y|\hat{x}), \log p(\hat{x})} BA_\beta$ it is a block matrix,

$$\begin{pmatrix} \frac{d \log p_{i+1}(y|\hat{x})}{d \log p_i(y'|\hat{x}')} & \frac{d \log p_{i+1}(y|\hat{x})}{d \log p_i(\hat{x})} \\ \frac{d \log p_{i+1}(\hat{x})}{d \log p_i(y'|\hat{x}')} & \frac{d \log p_{i+1}(\hat{x})}{d \log p_i(\hat{x})} \end{pmatrix} \tag{A25}$$

Its rows correspond to the output coordinates of BA_β . We index its upper rows by $y \in \mathcal{Y}$ and $\hat{x} \in \{1, \dots, T\}$, while its lower rows are indexed by \hat{x} alone. Similarly, its columns correspond to the input coordinates of BA_β . We index its leftmost columns by y' and \hat{x}' , and its rightmost columns by \hat{x}' alone. Each block in (A25) consists of contributions along all the distinct paths connecting two vertices in the dependencies graph (A6). For example, the lower-left block in (A25) consists of the contributions along all the paths in (A6) connecting $p_i(y'|\hat{x}')$ to $p_{i+1}(\hat{x})$.

We now spell out the paths contributing to each block in (A25), with repeated dummy indices understood to be summed over. Afterward, we shall calculate the contributing paths explicitly, carrying out the summations. The upper-left block of (A25) consists of

$$\frac{d \log p_{i+1}(y|\hat{x})}{d \log p_i(y'|\hat{x}')} = \frac{\partial \log p_{i+1}(y|\hat{x})}{\partial \log p_{i+1}(x_1|\hat{x}_2)} \cdot \left[\frac{\partial \log p_{i+1}(x_1|\hat{x}_2)}{\partial \log p_{i+1}(\hat{x}_3)} \frac{\partial \log p_{i+1}(\hat{x}_3)}{\partial \log p_{i+1}(\hat{x}_4|x_5)} + \frac{\partial \log p_{i+1}(x_1|\hat{x}_2)}{\partial \log p_{i+1}(\hat{x}_4|x_5)} \right] \cdot \left[\frac{\partial \log p_{i+1}(\hat{x}_4|x_5)}{\partial \log p_i(y'|\hat{x}')} + \frac{\partial \log p_{i+1}(\hat{x}_4|x_5)}{\partial \log Z_i(x_6, \beta)} \frac{\partial \log Z_i(x_6, \beta)}{\partial \log p_i(y'|\hat{x}')} \right] \tag{A26}$$

This Equation (A26) encodes the four paths connecting the vertex $p_i(y'|\hat{x}')$ to $p_{i+1}(y|\hat{x})$ in (A6). When accumulating the contributions in (A26), one must carefully sum only over repeated dummy indices that appear in the given term. For example, the two paths in (A26)

which traverse the edge $\frac{\partial \log p_{i+1}(x_1|\hat{x}_2)}{\partial \log p_{i+1}(\hat{x}_4|x_5)}$ (pointing from $p_{i+1}(\hat{x}|x)$ to $p_{i+1}(x|\hat{x})$) do not involve a summation over \hat{x}_3 . In contrast, the two paths involving $\frac{\partial \log p_{i+1}(x_1|\hat{x}_2)}{\partial \log p_{i+1}(\hat{x}_3)} \frac{\partial \log p_{i+1}(\hat{x}_3)}{\partial \log p_{i+1}(\hat{x}_4|x_5)}$ there do entail a summation over \hat{x}_3 . This is relevant for the calculations below, as in (A31) for example.

Similarly, for the upper-right block of (A25),

$$\begin{aligned} \frac{d \log p_{i+1}(y|\hat{x})}{d \log p_i(\hat{x}')} &= \frac{\partial \log p_{i+1}(y|\hat{x})}{\partial \log p_{i+1}(x_1|\hat{x}_2)} \cdot \left[\frac{\partial \log p_{i+1}(x_1|\hat{x}_2)}{\partial \log p_{i+1}(\hat{x}_3)} \frac{\partial \log p_{i+1}(\hat{x}_3)}{\partial \log p_{i+1}(\hat{x}_4|x_5)} + \frac{\partial \log p_{i+1}(x_1|\hat{x}_2)}{\partial \log p_{i+1}(\hat{x}_4|x_5)} \right] \\ &\cdot \left\{ \frac{\partial \log p_{i+1}(\hat{x}_4|x_5)}{\partial \log p_i(\hat{x}')} + \frac{\partial \log p_{i+1}(\hat{x}_4|x_5)}{\partial \log p_i(y_7|\hat{x}_8)} \frac{\partial \log p_i(y_7|\hat{x}_8)}{\partial \log p_i(x_9|\hat{x}_{10})} \frac{\partial \log p_i(x_9|\hat{x}_{10})}{\partial \log p_i(\hat{x}')} \right. \\ &\left. + \frac{\partial \log p_{i+1}(\hat{x}_4|x_5)}{\partial \log Z_i(x_6, \beta)} \left[\frac{\partial \log Z_i(x_6, \beta)}{\partial \log p_i(\hat{x}')} + \frac{\partial \log Z_i(x_6, \beta)}{\partial \log p_i(y_7|\hat{x}_8)} \frac{\partial \log p_i(y_7|\hat{x}_8)}{\partial \log p_i(x_9|\hat{x}_{10})} \frac{\partial \log p_i(x_9|\hat{x}_{10})}{\partial \log p_i(\hat{x}')} \right] \right\} \end{aligned} \tag{A27}$$

For the lower-left block of (A25),

$$\frac{d \log p_{i+1}(\hat{x})}{d \log p_i(y'|\hat{x}')} = \frac{\partial \log p_{i+1}(\hat{x})}{\partial \log p_{i+1}(\hat{x}_1|x_2)} \left[\frac{\partial \log p_{i+1}(\hat{x}_1|x_2)}{\partial \log p_i(y'|\hat{x}')} + \frac{\partial \log p_{i+1}(\hat{x}_1|x_2)}{\partial \log Z_i(x_3, \beta)} \frac{\partial \log Z_i(x_3, \beta)}{\partial \log p_i(y'|\hat{x}')} \right] \tag{A28}$$

Last, for the lower-right block of (A25),

$$\begin{aligned} \frac{d \log p_{i+1}(\hat{x})}{d \log p_i(\hat{x}')} &= \frac{\partial \log p_{i+1}(\hat{x})}{\partial \log p_{i+1}(\hat{x}_1|x_2)} \cdot \left\{ \frac{\partial \log p_{i+1}(\hat{x}_1|x_2)}{\partial \log p_i(\hat{x}')} + \frac{\partial \log p_{i+1}(\hat{x}_1|x_2)}{\partial \log p_i(y_3|\hat{x}_4)} \frac{\partial \log p_i(y_3|\hat{x}_4)}{\partial \log p_i(x_5|\hat{x}_6)} \frac{\partial \log p_i(x_5|\hat{x}_6)}{\partial \log p_i(\hat{x}')} \right. \\ &\left. + \frac{\partial \log p_{i+1}(\hat{x}_1|x_2)}{\partial \log Z_i(x_7, \beta)} \left[\frac{\partial \log Z_i(x_7, \beta)}{\partial \log p_i(\hat{x}')} + \frac{\partial \log Z_i(x_7, \beta)}{\partial \log p_i(y_3|\hat{x}_4)} \frac{\partial \log p_i(y_3|\hat{x}_4)}{\partial \log p_i(x_5|\hat{x}_6)} \frac{\partial \log p_i(x_5|\hat{x}_6)}{\partial \log p_i(\hat{x}')} \right] \right\} \end{aligned} \tag{A29}$$

Next, by using the intermediate results summarized in (A24) (Appendix B.1.1), we calculate each of the four blocks of (A25) explicitly. For the upper-left block (A26), we have

$$\begin{aligned} \frac{d \log p_{i+1}(y|\hat{x})}{d \log p_i(y'|\hat{x}')} &= \frac{p(y|x_1)p_{i+1}(x_1|\hat{x})}{p_{i+1}(y|\hat{x})} \delta_{\hat{x}, \hat{x}_2} \cdot [(-\delta_{\hat{x}_2, \hat{x}_3})p_{i+1}(x_5|\hat{x}_3)\delta_{\hat{x}_3, \hat{x}_4} + \delta_{x_1, x_5}\delta_{\hat{x}_2, \hat{x}_4}] \\ &\cdot [\beta\delta_{\hat{x}_4, \hat{x}'}p(y'|x_5) + (-\delta_{x_5, x_6})\beta p_{i+1}(\hat{x}'|x_6)p(y'|x_6)] \end{aligned} \tag{A30}$$

For clarity, we elaborate on each step needed to complete the calculation of the upper-left block (A26) while providing only the main steps for the other blocks. To carry out the summations over the dummy variables $x_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, x_5,$ and x_6 in (A30), we carefully sum only over repeated dummy indices, as explained after (A26). We carry out one summation at a time, starting with \hat{x}_2 . This yields

$$\begin{aligned} &\beta \frac{p(y|x_1)p_{i+1}(x_1|\hat{x})}{p_{i+1}(y|\hat{x})} \cdot [-\delta_{\hat{x}, \hat{x}_3}p_{i+1}(x_5|\hat{x}_3)\delta_{\hat{x}_3, \hat{x}_4} + \delta_{x_1, x_5}\delta_{\hat{x}_2, \hat{x}_4}] \cdot [\delta_{\hat{x}_4, \hat{x}'}p(y'|x_5) - \delta_{x_5, x_6} p_{i+1}(\hat{x}'|x_6)p(y'|x_6)] \\ &= \beta \cdot \left[-\delta_{\hat{x}, \hat{x}_3}p_{i+1}(x_5|\hat{x}_3)\delta_{\hat{x}_3, \hat{x}_4} + \delta_{\hat{x}, \hat{x}_4} \frac{p(y|x_5)p_{i+1}(x_5|\hat{x})}{p_{i+1}(y|\hat{x})} \right] \cdot [\delta_{\hat{x}_4, \hat{x}'}p(y'|x_5) - \delta_{x_5, x_6} p_{i+1}(\hat{x}'|x_6)p(y'|x_6)] \\ &= \beta \cdot p_{i+1}(x_5|\hat{x}) \left[-\delta_{\hat{x}, \hat{x}_4} + \delta_{\hat{x}, \hat{x}_4} \frac{p(y|x_5)}{p_{i+1}(y|\hat{x})} \right] \cdot [\delta_{\hat{x}_4, \hat{x}'}p(y'|x_5) - \delta_{x_5, x_6} p_{i+1}(\hat{x}'|x_6)p(y'|x_6)] \\ &= -\beta \cdot p_{i+1}(x_5|\hat{x}) \left[1 - \frac{p(y|x_5)}{p_{i+1}(y|\hat{x})} \right] \cdot [\delta_{\hat{x}, \hat{x}'}p(y'|x_5) - \delta_{x_5, x_6} p_{i+1}(\hat{x}'|x_6)p(y'|x_6)] \\ &= -\beta \cdot p(y'|x_5)p_{i+1}(x_5|\hat{x}) \left[1 - \frac{p(y|x_5)}{p_{i+1}(y|\hat{x})} \right] \cdot [\delta_{\hat{x}, \hat{x}'} - p_{i+1}(\hat{x}'|x_5)] \\ &= -\beta \sum_x p(y'|x)p_{i+1}(x|\hat{x}) \cdot \left[1 - \frac{p(y|x)}{p_{i+1}(y|\hat{x})} \right] \cdot [\delta_{\hat{x}, \hat{x}'} - p_{i+1}(\hat{x}'|x)] \end{aligned} \tag{A31}$$

In the first equality above we carried out the summation over x_1 , in the second over \hat{x}_3 , in the third over \hat{x}_4 , in the fourth over x_6 , and in the fifth over x_5 .

To simplify the notation, we replace summations over x with definitions as in Equation (14) (Section 3),

$$\begin{aligned}
 C(\hat{x}, \hat{x}'; i)_{y,y'} &:= \sum_x p(y|x)p(y'|x)p_i(\hat{x}'|x)p_i(x|\hat{x}) \\
 B(\hat{x}, \hat{x}'; i)_y &:= \sum_x p(y|x)p_i(\hat{x}'|x)p_i(x|\hat{x}) = \sum_{y'} C(\hat{x}, \hat{x}'; i)_{y,y'} \\
 A(\hat{x}, \hat{x}'; i) &:= \sum_x p_i(\hat{x}'|x)p_i(x|\hat{x}) = \sum_y B(\hat{x}, \hat{x}'; i)_y \\
 D(\hat{x}; i)_{y,y'} &:= \frac{1}{p_i(y|\hat{x})} \sum_x p(y|x)p(y'|x)p_i(x|\hat{x}) = \frac{1}{p_i(y|\hat{x})} \sum_{\hat{x}'} C(\hat{x}, \hat{x}'; i)_{y,y'}
 \end{aligned}
 \tag{A32}$$

and note that

$$\sum_{y', \hat{x}'} C(\hat{x}, \hat{x}'; i)_{y,y'} = p_i(y|\hat{x}). \tag{A33}$$

The quantities A, B , and C involve two IB clusters. They are a scalar, a vector, and a matrix, respectively. The definition of D involves only one IB cluster and coincides with C_Y in [13] (3.2 in Part III). The relations to the right of (A32) show that each can be expressed in terms of $C(\hat{x}, \hat{x}'; i)_{y,y'}$. Equation (A33) shows that the latter can be rewritten as a right-stochastic matrix, up to trivial manipulations. As seen below, the Jacobian matrix (A25) of a BA-IB step in log-decoder coordinates can be computed in terms of the quantities in (A32).

With the latter definitions (A32), (A31) can be rewritten as

$$\begin{aligned}
 \frac{d \log p_{i+1}(y|\hat{x})}{d \log p_i(y'|\hat{x}')} &\stackrel{(A31)}{=} -\beta \sum_x \left[\delta_{\hat{x}, \hat{x}'} p(y'|x)p_{i+1}(x|\hat{x}) - p(y'|x)p_{i+1}(\hat{x}'|x)p_{i+1}(x|\hat{x}) \right. \\
 &\quad \left. - \delta_{\hat{x}, \hat{x}'} \frac{1}{p_{i+1}(y|\hat{x})} p(y|x)p(y'|x)p_{i+1}(x|\hat{x}) + \frac{1}{p_{i+1}(y|\hat{x})} p(y'|x)p(y|x)p_{i+1}(\hat{x}'|x)p_{i+1}(x|\hat{x}) \right] \\
 &\stackrel{(A32)}{=} -\beta \left[\delta_{\hat{x}, \hat{x}'} p_{i+1}(y'|\hat{x}) - B(\hat{x}, \hat{x}'; i+1)_{y'} - \delta_{\hat{x}, \hat{x}'} D(\hat{x}; i+1)_{y,y'} + \frac{1}{p_{i+1}(y|\hat{x})} C(\hat{x}, \hat{x}'; i+1)_{y,y'} \right] \\
 &= \beta \sum_{\hat{x}'', y''} (\delta_{\hat{x}'', \hat{x}'} - \delta_{\hat{x}, \hat{x}''}) \left(1 - \frac{\delta_{y'', y}}{p_{i+1}(y|\hat{x})} \right) C(\hat{x}, \hat{x}''; i+1)_{y'', y''}
 \end{aligned}
 \tag{A34}$$

The third equality above follows from (A33), the identities to the right of (A32), and simple algebra.

For the upper-right block (A27),

$$\begin{aligned}
 \frac{d \log p_{i+1}(y|\hat{x})}{d \log p_i(\hat{x}')} &= \frac{p(y|x_1)p_{i+1}(x_1|\hat{x}_2)}{p_{i+1}(y|\hat{x}_2)} \delta_{\hat{x}_2, \hat{x}} \cdot \left[(-\delta_{\hat{x}_2, \hat{x}_3}) p_{i+1}(x_5|\hat{x}_3) \delta_{\hat{x}_3, \hat{x}_4} + \delta_{x_1, x_5} \delta_{\hat{x}_2, \hat{x}_4} \right] \\
 &\quad \cdot \left\{ \delta_{\hat{x}_4, \hat{x}'} + \beta \delta_{\hat{x}_4, \hat{x}_8} p(y_7|x_5) \frac{p(y_7|x_9)p_i(x_9|\hat{x}_8)}{p_i(y_7|\hat{x}_8)} \delta_{\hat{x}_8, \hat{x}_{10}} (-\delta_{\hat{x}_{10}, \hat{x}'} \right. \\
 &\quad \left. + (-\delta_{x_5, x_6}) \left[p_{i+1}(\hat{x}'|x_6) + \beta p_{i+1}(\hat{x}_8|x_6) p(y_7|x_6) \frac{p(y_7|x_9)p_i(x_9|\hat{x}_8)}{p_i(y_7|\hat{x}_8)} \delta_{\hat{x}_8, \hat{x}_{10}} (-\delta_{\hat{x}_{10}, \hat{x}'} \right] \right\}
 \end{aligned}
 \tag{A35}$$

In a manner similar to (A31), summing over all ten dummy variables other than x_1 and x_5 yields

$$\begin{aligned}
 &(1 - \beta) \cdot \frac{p(y|x_1)p_{i+1}(x_1|\hat{x})}{p_{i+1}(y|\hat{x})} \cdot (\delta_{x_1, x_5} - p_{i+1}(x_5|\hat{x})) \cdot (\delta_{\hat{x}, \hat{x}'} - p_{i+1}(\hat{x}'|x_5)) \\
 &= (1 - \beta) \cdot \left(\frac{-1}{p_{i+1}(y|\hat{x})} \sum_x p(y|x)p_{i+1}(\hat{x}'|x)p_{i+1}(x|\hat{x}) + \sum_x p_{i+1}(\hat{x}'|x)p_{i+1}(x|\hat{x}) \right) \\
 &= (1 - \beta) \cdot \sum_x \left(1 - \frac{p(y|x)}{p_{i+1}(y|\hat{x})} \right) p_{i+1}(\hat{x}'|x)p_{i+1}(x|\hat{x})
 \end{aligned}
 \tag{A36}$$

The two terms involving $\delta_{\hat{x}, \hat{x}'}$ cancel out when summing over x_1 and x_5 at the first equality. Rewriting with the definitions (A32) of A and B further simplifies (A36) to

$$(1 - \beta) \cdot \left[A(\hat{x}, \hat{x}'; i + 1) - \frac{1}{p_{i+1}(y|\hat{x})} B(\hat{x}, \hat{x}'; i + 1)_y \right] = (1 - \beta) \cdot \sum_{y''} \left[1 - \frac{\delta_{y'', y}}{p_{i+1}(y|\hat{x})} \right] B(\hat{x}, \hat{x}'; i + 1)_{y''} \tag{A37}$$

For the lower-left block (A28),

$$\frac{d \log p_{i+1}(\hat{x})}{d \log p_i(y'|\hat{x}')} = p_{i+1}(x_2|\hat{x}) \delta_{\hat{x}, \hat{x}'} [\beta \delta_{\hat{x}_1, \hat{x}'} p(y'|x_2) + (-\delta_{x_2, x_3}) \beta p_{i+1}(\hat{x}'|x_3) p(y'|x_3)] \tag{A38}$$

Summing over dummy variables and simplifying yields

$$\beta \cdot \left[\delta_{\hat{x}, \hat{x}'} p_{i+1}(y'|\hat{x}) - \sum_x p(y'|x) p_{i+1}(\hat{x}'|x) p_{i+1}(x|\hat{x}) \right] \tag{A39}$$

In terms of definitions (A32), this simplifies to

$$\beta \cdot \left[\delta_{\hat{x}, \hat{x}'} p_{i+1}(y'|\hat{x}) - B(\hat{x}, \hat{x}'; i + 1)_{y'} \right] \tag{A40}$$

Finally, for the lower-right block (A29),

$$\frac{d \log p_{i+1}(\hat{x})}{d \log p_i(\hat{x}')} = p_{i+1}(x_2|\hat{x}) \delta_{\hat{x}, \hat{x}'} \cdot \left\{ \delta_{\hat{x}_1, \hat{x}'} + \beta \delta_{\hat{x}_1, \hat{x}_4} p(y_3|x_2) \frac{p(y_3|x_5) p_i(x_5|\hat{x}_4)}{p_i(y_3|\hat{x}_4)} \delta_{\hat{x}_4, \hat{x}_6} (-\delta_{\hat{x}_6, \hat{x}'}) + (-\delta_{x_2, x_7}) \left[p_{i+1}(\hat{x}'|x_7) + \beta p_{i+1}(\hat{x}_4|x_7) p(y_3|x_7) \frac{p(y_3|x_5) p_i(x_5|\hat{x}_4)}{p_i(y_3|\hat{x}_4)} \delta_{\hat{x}_4, \hat{x}_6} (-\delta_{\hat{x}_6, \hat{x}'}) \right] \right\} \tag{A41}$$

This simplifies to

$$(1 - \beta) \left(\delta_{\hat{x}, \hat{x}'} - \sum_x p_{i+1}(\hat{x}'|x) p_{i+1}(x|\hat{x}) \right) \tag{A42}$$

With definitions (A32), this can be written as

$$(1 - \beta) \left(\delta_{\hat{x}, \hat{x}'} - A(\hat{x}, \hat{x}'; i + 1) \right) \tag{A43}$$

Collecting the results from (A34), (A37), (A40), and (A43) back into (A25), BA's Jacobian in these coordinates is

$$\left(\begin{array}{c|c} \beta \sum_{\hat{x}'', y''} (\delta_{\hat{x}'', \hat{x}'} - \delta_{\hat{x}, \hat{x}'}) \cdot \left(1 - \frac{\delta_{y'', y}}{p_{i+1}(y|\hat{x})} \right) C(\hat{x}, \hat{x}'', i+1)_{y', y''} & (1 - \beta) \cdot \sum_{y''} \left[1 - \frac{\delta_{y'', y}}{p_{i+1}(y|\hat{x})} \right] B(\hat{x}, \hat{x}'; i + 1)_{y''} \\ \hline \beta \cdot \left[\delta_{\hat{x}, \hat{x}'} p_{i+1}(y'|\hat{x}) - B(\hat{x}, \hat{x}'; i + 1)_{y'} \right] & (1 - \beta) \left(\delta_{\hat{x}, \hat{x}'} - A(\hat{x}, \hat{x}'; i + 1) \right) \end{array} \right) \tag{A44}$$

When evaluated at an IB root, this is Equation (13) of Section 3. Equivalently, it can be written in the following form, which is more convenient for implementation:

$$\left(\begin{array}{c|c} \beta \left[B(\hat{x}, \hat{x}'; i+1)_{y'} - \delta_{\hat{x}, \hat{x}'} p_{i+1}(y'|\hat{x}) \right. \\ \left. + \delta_{\hat{x}, \hat{x}'} D(\hat{x}; i+1)_{y, y'} - \frac{1}{p_{i+1}(y|\hat{x})} C(\hat{x}, \hat{x}'; i+1)_{y, y'} \right] & (1 - \beta) \cdot \left[A(\hat{x}, \hat{x}'; i+1) - \frac{1}{p_{i+1}(y|\hat{x})} B(\hat{x}, \hat{x}'; i+1)_y \right] \\ \hline \beta \cdot \left[\delta_{\hat{x}, \hat{x}'} p_{i+1}(y'|\hat{x}) - B(\hat{x}, \hat{x}'; i+1)_{y'} \right] & (1 - \beta) \left(\delta_{\hat{x}, \hat{x}'} - A(\hat{x}, \hat{x}'; i+1) \right) \end{array} \right) \quad (A45)$$

Appendix B.3. The Partial β -Derivatives of BA-IB in Log-Decoder Coordinates

We calculate the vector $D_\beta BA_\beta$ of partial derivatives of the BA_β operator in log-decoder coordinates (of Section 2), which appears at the right-hand side of the IB-ODE (16) (in Section 3).

To that end, we differentiate backward along the dependencies graph (A6) (in Appendix B.1) with respect to β , starting at the output coordinates $p_{i+1}(y|\hat{x})$ and $p_{i+1}(\hat{x})$ of BA_β . After differentiating, we mind our independent variables. Here, these are β , and the input coordinates $p_i(y|\hat{x})$ and $p_i(\hat{x})$ of BA_β . The differentiation of these with respect to β vanishes (except for $\frac{d\beta}{d\beta} = 1$), as they are independent. Finally, we compose the differentiations to obtain the effect $D_\beta BA_\beta$ of changing β on BA's output. We note that, in principle, one can differentiate the explicit formulae (A1) of BA_β in decoder coordinates (Appendix A) with respect to β . However, we find that to be cumbersome and far more error-prone than our approach, and so proceed in the spirit of the previous Appendix B.2.

We start by differentiating each of the equations defining the Blahut–Arimoto Algorithm 1 with respect to β , as if all its variables are dependent. For the cluster marginal Step 1.4,

$$\frac{d}{d\beta} p_i(\hat{x}) = \sum_x p(x) \frac{d}{d\beta} p_i(\hat{x}|x) \quad (A46)$$

For the inverse encoder Step 1.5,

$$\frac{d}{d\beta} p_i(x|\hat{x}) = \frac{p(x)}{p_i(\hat{x})} \frac{dp_i(\hat{x}|x)}{d\beta} - \frac{p_i(\hat{x}|x)p(x)}{p_i(\hat{x})^2} \frac{dp_i(\hat{x})}{d\beta} \quad (A47)$$

For the decoder Step 1.6,

$$\frac{d}{d\beta} p_i(y|\hat{x}) = \sum_x p(y|x) \frac{d}{d\beta} p_i(x|\hat{x}) \quad (A48)$$

For the KL-divergence,

$$\frac{d}{d\beta} D_{KL}[p(y|x)||p_i(y|\hat{x})] = \frac{d}{d\beta} \sum_{y''} p(y''|x) \log \frac{p(y''|x)}{p_i(y''|\hat{x})} = - \sum_{y''} \frac{p(y''|x)}{p_i(y''|\hat{x})} \frac{d}{d\beta} p_i(y''|\hat{x}) \quad (A49)$$

And for its exponent,

$$\begin{aligned} \frac{d}{d\beta} \exp \{ -\beta D_{x, \hat{x}} \} &= - \left(D_{x, \hat{x}} + \beta \frac{dD_{x, \hat{x}}}{d\beta} \right) \cdot \exp \{ -\beta D_{x, \hat{x}} \} \\ &\stackrel{(A49)}{=} - \left(D_{x, \hat{x}} - \beta \sum_{y''} \frac{p(y''|x)}{p_i(y''|\hat{x})} \frac{d}{d\beta} p_i(y''|\hat{x}) \right) \cdot \exp \{ -\beta D_{x, \hat{x}} \} \end{aligned} \quad (A50)$$

where we have written $D_{x, \hat{x}} := D_{KL}[p(y|x)||p_i(y|\hat{x})]$ for short. Thus, for the partition function's Step 1.7, we have

$$\begin{aligned} \frac{d}{d\beta} Z_i(x, \beta) &= \frac{d}{d\beta} \sum_{\hat{x}''} p_i(\hat{x}'') \exp \{ -\beta D_{x, \hat{x}''} \} \\ &\stackrel{(A50)}{=} \sum_{\hat{x}''} \left(\frac{dp_i(\hat{x}'')}{d\beta} - p_i(\hat{x}'') D_{x, \hat{x}''} + \beta p_i(\hat{x}'') \sum_{y''} \frac{p(y''|x)}{p_i(y''|\hat{x}'')} \frac{d}{d\beta} p_i(y''|\hat{x}'') \right) \cdot \exp \{ -\beta D_{x, \hat{x}''} \} \end{aligned} \quad (A51)$$

Finally, for the encoder Step 1.8 we have

$$\begin{aligned} \frac{d}{d\beta} p_{i+1}(\hat{x}|x) &= \frac{d}{d\beta} \left(\frac{p_i(\hat{x}) e^{-\beta D_{x, \hat{x}}}}{Z_i(x, \beta)} \right) \\ &\stackrel{(A50)}{=} \frac{p_i(\hat{x}) e^{-\beta D_{x, \hat{x}}}}{Z_i(x, \beta)} \left[\frac{1}{p_i(\hat{x})} \frac{dp_i(\hat{x})}{d\beta} - \left(D_{x, \hat{x}} - \beta \sum_{y''} \frac{p(y''|x)}{p_i(y''|\hat{x})} \frac{d}{d\beta} p_i(y''|\hat{x}) \right) - \frac{1}{Z_i(x, \beta)} \frac{dZ_i(x, \beta)}{d\beta} \right] \\ &\stackrel{\text{Step 1.8}}{=} p_{i+1}(\hat{x}|x) \cdot \left[\frac{1}{p_i(\hat{x})} \frac{dp_i(\hat{x})}{d\beta} - \left(D_{x, \hat{x}} - \beta \sum_{y''} \frac{p(y''|x)}{p_i(y''|\hat{x})} \frac{d}{d\beta} p_i(y''|\hat{x}) \right) - \frac{1}{Z_i(x, \beta)} \frac{dZ_i(x, \beta)}{d\beta} \right] \end{aligned} \quad (A52)$$

Next, picking β and the inputs $\log p_i(y|\hat{x})$ and $\log p_i(\hat{x})$ of BA_β as our independent variables, we compose the differentiations above to obtain $D_\beta BA_\beta$ at an output coordinate. That is, we seek $\frac{d}{d\beta} \log p_{i+1}(y|\hat{x})$ and $\frac{d}{d\beta} \log p_{i+1}(\hat{x})$. By the chain rule, we trace the dependencies graph (A6) (Appendix B.1) backwards, from the output nodes $p_{i+1}(y|\hat{x})$ and $p_{i+1}(\hat{x})$ back to the input nodes. The derivatives of the latter with respect to β vanish, as these are our independent variables.

Starting with a decoder output coordinate,

$$\begin{aligned} \frac{d}{d\beta} \log p_{i+1}(y|\hat{x}) &= \frac{1}{p_{i+1}(y|\hat{x})} \frac{d}{d\beta} p_{i+1}(y|\hat{x}) \stackrel{(A48)}{=} \frac{1}{p_{i+1}(y|\hat{x})} \sum_x p(y|x) \frac{d}{d\beta} p_{i+1}(x|\hat{x}) \\ &\stackrel{(A47)}{=} \frac{1}{p_{i+1}(y|\hat{x})} \sum_x p(y|x) \left[\frac{p(x)}{p_{i+1}(\hat{x})} \frac{dp_{i+1}(\hat{x}|x)}{d\beta} - \frac{p_{i+1}(\hat{x}|x)p(x)}{p_{i+1}(\hat{x})^2} \frac{dp_{i+1}(\hat{x})}{d\beta} \right] \\ &\stackrel{(A46)}{=} \sum_x \frac{p(y|x)p(x)}{p_{i+1}(y|\hat{x})p_{i+1}(\hat{x})} \left[\frac{dp_{i+1}(\hat{x}|x)}{d\beta} - \frac{p_{i+1}(\hat{x}|x)}{p_{i+1}(\hat{x})} \sum_{x'} p(x') \frac{dp_{i+1}(\hat{x}|x')}{d\beta} \right] \\ &\stackrel{(A52)}{=} \sum_x \frac{p(y|x)p(x)}{p_{i+1}(y|\hat{x})p_{i+1}(\hat{x})} \left\{ p_{i+1}(\hat{x}|x) \cdot \left[\frac{1}{p_i(\hat{x})} \frac{dp_i(\hat{x})}{d\beta} - \left(D_{x, \hat{x}} - \beta \sum_{y''} \frac{p(y''|x)}{p_i(y''|\hat{x})} \frac{dp_i(y''|\hat{x})}{d\beta} \right) - \frac{1}{Z_i(x, \beta)} \frac{dZ_i(x, \beta)}{d\beta} \right] \right. \\ &\quad \left. - \frac{p_{i+1}(\hat{x}|x)}{p_{i+1}(\hat{x})} \sum_{x'} p(x') \left(p_{i+1}(\hat{x}|x') \cdot \left[\frac{1}{p_i(\hat{x})} \frac{dp_i(\hat{x})}{d\beta} - \left(D_{x', \hat{x}} - \beta \sum_{y''} \frac{p(y''|x')}{p_i(y''|\hat{x})} \frac{dp_i(y''|\hat{x})}{d\beta} \right) - \frac{1}{Z_i(x', \beta)} \frac{dZ_i(x', \beta)}{d\beta} \right] \right) \right\} \end{aligned} \quad (A53)$$

Since $p_i(y|\hat{x})$ and $p_i(\hat{x})$ are independent input variables, their derivatives with respect to the independent variable β vanish, yielding

$$\begin{aligned} & - \sum_x \frac{p(y|x)p(x)}{p_{i+1}(y|\hat{x})p_{i+1}(\hat{x})} \left\{ p_{i+1}(\hat{x}|x) \cdot \left[D_{x, \hat{x}} + \frac{1}{Z_i(x, \beta)} \frac{dZ_i(x, \beta)}{d\beta} \right] \right. \\ & \quad \left. - \frac{p_{i+1}(\hat{x}|x)}{p_{i+1}(\hat{x})} \sum_{x'} p(x') p_{i+1}(\hat{x}|x') \cdot \left[D_{x', \hat{x}} + \frac{1}{Z_i(x', \beta)} \frac{dZ_i(x', \beta)}{d\beta} \right] \right\} \end{aligned} \quad (A54)$$

To complete the calculation at (A53), note that the same argument can be used for two of the three summands in (A51), reducing it to

$$\frac{dZ_i(x, \beta)}{d\beta} = - \sum_{\hat{x}''} p_i(\hat{x}'') D_{x, \hat{x}''} e^{-\beta D_{x, \hat{x}''}} \quad (A55)$$

since $p_i(y|\hat{x})$ and $p_i(\hat{x})$ are considered as independent variables. Therefore,

$$\begin{aligned}
 \frac{d}{d\beta} \log p_{i+1}(y|\hat{x}) &\stackrel{(A54)}{=} - \sum_x \frac{p(y|x)p(x)}{p_{i+1}(y|\hat{x})p_{i+1}(\hat{x})} \left\{ p_{i+1}(\hat{x}|x) \cdot \left[D_{x,\hat{x}} - \sum_{\hat{x}''} \left(\frac{p_i(\hat{x}'')}{Z_i(x,\beta)} e^{-\beta D_{x,\hat{x}''}} \right) D_{x,\hat{x}''} \right] \right. \\
 &\quad \left. - \frac{p_{i+1}(\hat{x}|x)}{p_{i+1}(\hat{x})} \sum_{x'} p(x') p_{i+1}(\hat{x}|x') \cdot \left[D_{x',\hat{x}} - \sum_{\hat{x}''} \left(\frac{p_i(\hat{x}'')}{Z_i(x',\beta)} e^{-\beta D_{x',\hat{x}''}} \right) D_{x',\hat{x}''} \right] \right\} \\
 &\stackrel{\text{Step 1.8}}{=} - \sum_x \frac{p(y|x)p(x)}{p_{i+1}(y|\hat{x})p_{i+1}(\hat{x})} \left\{ p_{i+1}(\hat{x}|x) \cdot \left[D_{x,\hat{x}} - \sum_{\hat{x}''} p_{i+1}(\hat{x}''|x) D_{x,\hat{x}''} \right] \right. \\
 &\quad \left. - \frac{p_{i+1}(\hat{x}|x)}{p_{i+1}(\hat{x})} \sum_{x'} p(x') p_{i+1}(\hat{x}|x') \cdot \left[D_{x',\hat{x}} - \sum_{\hat{x}''} p_{i+1}(\hat{x}''|x') D_{x',\hat{x}''} \right] \right\} \\
 &\stackrel{\text{Step 1.5}}{\stackrel{\text{Step 1.6}}{=}} \sum_x p_{i+1}(x|\hat{x}) D_{x,\hat{x}} - \sum_x \frac{p(y|x)}{p_{i+1}(y|\hat{x})} p_{i+1}(x|\hat{x}) D_{x,\hat{x}} \\
 &\quad + \sum_{x,\hat{x}''} \frac{p(y|x)}{p_{i+1}(y|\hat{x})} p_{i+1}(\hat{x}''|x) p_{i+1}(x|\hat{x}) D_{x,\hat{x}''} - \sum_{x,\hat{x}''} p_{i+1}(\hat{x}''|x) p_{i+1}(x|\hat{x}) D_{x,\hat{x}''} \\
 &= \sum_x \left[1 - \frac{p(y|x)}{p_{i+1}(y|\hat{x})} \right] p_{i+1}(x|\hat{x}) D_{x,\hat{x}} - \sum_{x,\hat{x}''} \left[1 - \frac{p(y|x)}{p_{i+1}(y|\hat{x})} \right] p_{i+1}(\hat{x}''|x) p_{i+1}(x|\hat{x}) D_{x,\hat{x}''} \tag{A56}
 \end{aligned}$$

At the second equality to the bottom we started with the third summand, then with the first, and only then with the third and fourth summands. And so,

$$\frac{d}{d\beta} \log p_{i+1}(y|\hat{x}) = \sum_{x,\hat{x}''} \left[1 - \frac{p(y|x)}{p_{i+1}(y|\hat{x})} \right] \cdot \left[\delta_{\hat{x},\hat{x}''} - p_{i+1}(\hat{x}''|x) \right] \cdot p_{i+1}(x|\hat{x}) D_{x,\hat{x}''} \tag{A57}$$

Next, consider a cluster marginal output coordinate,

$$\begin{aligned}
 \frac{d}{d\beta} \log p_{i+1}(\hat{x}) &= \frac{1}{p_{i+1}(\hat{x})} \frac{d}{d\beta} p_{i+1}(\hat{x}) \stackrel{(A46)}{=} \frac{1}{p_{i+1}(\hat{x})} \sum_x p(x) \frac{d}{d\beta} p_{i+1}(\hat{x}|x) \\
 &\stackrel{(A52)}{=} \frac{1}{p_{i+1}(\hat{x})} \sum_x p(x) p_{i+1}(\hat{x}|x) \cdot \left[\frac{1}{p_i(\hat{x})} \frac{dp_i(\hat{x})}{d\beta} - \left(D_{x,\hat{x}} - \beta \sum_{y''} \frac{p(y''|x)}{p_i(y''|\hat{x})} \frac{d}{d\beta} p_i(y''|\hat{x}) \right) - \frac{1}{Z_i(x,\beta)} \frac{dZ_i(x,\beta)}{d\beta} \right] \tag{A58}
 \end{aligned}$$

Since $p_i(y|\hat{x})$ and $p_i(\hat{x})$ are independent variables, their derivatives with respect to β vanish, yielding

$$\begin{aligned}
 &- \frac{1}{p_{i+1}(\hat{x})} \sum_x p_{i+1}(\hat{x}|x) p(x) \left[D_{x,\hat{x}} + \frac{1}{Z_i(x,\beta)} \frac{dZ_i(x,\beta)}{d\beta} \right] \\
 &\stackrel{(A55)}{=} - \frac{1}{p_{i+1}(\hat{x})} \sum_x p_{i+1}(\hat{x}|x) p(x) \left[D_{x,\hat{x}} - \sum_{\hat{x}''} \left(\frac{p_i(\hat{x}'')}{Z_i(x,\beta)} e^{-\beta D_{x,\hat{x}''}} \right) D_{x,\hat{x}''} \right] \\
 &\stackrel{\text{Step 1.8}}{\stackrel{\text{Step 1.5}}{=}} - \sum_{x,\hat{x}''} \left[\delta_{\hat{x},\hat{x}''} - p_{i+1}(\hat{x}''|x) \right] \cdot p_{i+1}(x|\hat{x}) D_{x,\hat{x}''} \tag{A59}
 \end{aligned}$$

Thus, for the marginals' coordinates, we have obtained

$$\frac{d}{d\beta} \log p_{i+1}(\hat{x}) = - \sum_{x,\hat{x}''} \left[\delta_{\hat{x},\hat{x}''} - p_{i+1}(\hat{x}''|x) \right] \cdot p_{i+1}(x|\hat{x}) D_{x,\hat{x}''} \tag{A60}$$

When evaluated at an IB root, Equations (A57) and (A60) form, respectively, the decoder and marginal coordinates of $D_\beta B A_\beta$, which appears at the right-hand side of the IB ODE (16) (note the extra minus sign in the implicit ODE (7)).

Appendix B.4. The Coordinate Exchange Jacobians between Log-Decoder and Log-Encoder Coordinates

Following the discussion in Section 2 on the pros and cons of each coordinate system, we leverage the observations of Appendix B.1 in order to derive the coordinate exchange Jacobians between the log-decoder and log-encoder coordinate systems. Exchanging between the other coordinate system pairs adds little to the below and thus is omitted.

Given the encoder’s logarithmic derivative $\frac{d}{d\beta} \log p_\beta(\hat{x}'|x')$, we would like to compute from it the logarithmic derivative $(\frac{d}{d\beta} \log p_\beta(y|\hat{x}), \frac{d}{d\beta} \log p_\beta(\hat{x}))$ in decoder coordinates, and vice versa. To that end, recall that an (arbitrary) encoder $p(\hat{x}'|\hat{x})$ determines a decoder–marginal pair $(p(y|\hat{x}), p(\hat{x}))$ and vice versa (e.g., Equation (11) in Section 2). So, one can follow the dependencies graph (A6) (in Appendix B.1) backward between these coordinate systems to exchange the coordinates of an implicit derivative. For example, consider $p_i(y|\hat{x})$ and $p_i(\hat{x})$ as functions of the encoder $p_i(\hat{x}'|x')$ preceding it in the graph (A6). When at an IB root, multiplying by the coordinates’ exchange Jacobian yields

$$\frac{d \log p_\beta(y|\hat{x})}{d\beta} = \frac{d \log p_\beta(y|\hat{x})}{d \log p_\beta(\hat{x}'|x')} \frac{d \log p_\beta(\hat{x}'|x')}{d\beta} \quad \text{and} \quad \text{(A61)}$$

$$\frac{d \log p_\beta(\hat{x})}{d\beta} = \frac{d \log p_\beta(\hat{x})}{d \log p_\beta(\hat{x}'|x')} \frac{d \log p_\beta(\hat{x}'|x')}{d\beta}. \quad \text{(A62)}$$

Similarly, considering an encoder $p_\beta(\hat{x}|x)$ as a function of $p_\beta(y'|\hat{x}')$ and $p_\beta(\hat{x}')$,

$$\frac{d \log p_\beta(\hat{x}|x)}{d\beta} = \frac{d \log p_\beta(\hat{x}|x)}{d \log p_\beta(y'|\hat{x}')} \frac{d \log p_\beta(y'|\hat{x}')}{d\beta} + \frac{d \log p_\beta(\hat{x}|x)}{d \log p_\beta(\hat{x}')} \frac{d \log p_\beta(\hat{x}')}{d\beta} + \frac{\partial \log p_\beta(\hat{x}|x)}{\partial \beta}. \quad \text{(A63)}$$

The last term $\frac{\partial \log p_\beta(\hat{x}|x)}{\partial \beta}$ in (A63) stems from the fact that the encoder Step 1.8 depends explicitly on β , unlike the marginal and decoder Steps 1.4 and 1.6. cf., the comments around (A8) in Appendix B.1.

The matrices $\frac{d \log p_\beta(y|\hat{x})}{d \log p_\beta(\hat{x}'|x')}$ and $\frac{d \log p_\beta(\hat{x})}{d \log p_\beta(\hat{x}'|x')}$ for exchanging from encoder to decoder coordinates follow from the chain rule, and are calculated in Appendix B.4.1 below, at Equations (A64) and (A66). Similarly, the matrices $\frac{d \log p_\beta(\hat{x}|x)}{d \log p_\beta(y'|\hat{x}')}$ and $\frac{d \log p_\beta(\hat{x}|x)}{d \log p_\beta(\hat{x}')}$ and the partial derivative $\frac{\partial \log p_\beta(\hat{x}|x)}{\partial \beta}$ for exchanging from decoder to encoder coordinates are Equations (A68), (A70), and (A73), in Appendix B.4.2.

Appendix B.4.1. Exchanging from Encoder to Decoder Coordinates

An input encoder $p_i(\hat{x}'|x')$ determines a decoder $p_i(y|\hat{x})$ and a marginal $p_i(\hat{x})$. As in previous subsections, we follow the dependencies graph (A6) along all the paths between these.

Using diagram (A24) from Appendix B.1.1, for the marginal one has

$$\frac{d \log p_i(\hat{x})}{d \log p_i(\hat{x}'|x')} = p_i(x'|x') \delta_{\hat{x}, \hat{x}'}, \quad \text{(A64)}$$

while for the decoder,

$$\begin{aligned} \frac{d \log p_i(y|\hat{x})}{d \log p_i(\hat{x}'|x')} &= \frac{\partial \log p_i(y|\hat{x})}{\partial \log p_i(x_1|\hat{x}_2)} \left[\frac{\partial \log p_i(x_1|\hat{x}_2)}{\partial \log p_i(\hat{x}'|x')} + \frac{\partial \log p_i(x_1|\hat{x}_2)}{\partial \log p_i(\hat{x}_3)} \frac{\partial \log p_i(\hat{x}_3)}{\partial \log p_i(\hat{x}'|x')} \right] \\ &= \frac{p(y|x_1)p_i(x_1|\hat{x}_2)}{p_i(y|\hat{x}_2)} \delta_{\hat{x}_2, \hat{x}} \left[\delta_{x_1, x'} \delta_{\hat{x}_2, \hat{x}'} - \delta_{\hat{x}_2, \hat{x}_3} p_i(x'|\hat{x}_3) \delta_{\hat{x}_3, \hat{x}'} \right] \end{aligned} \quad \text{(A65)}$$

Summing over the three dummy variables as before, the latter simplifies to

$$\frac{d \log p_i(y|\hat{x})}{d \log p_i(\hat{x}'|x')} = \left[\frac{p(y|x')}{p_i(y|\hat{x})} - 1 \right] p_i(x'|\hat{x}) \delta_{\hat{x},x'} . \tag{A66}$$

Appendix B.4.2. Exchanging from Decoder to Encoder Coordinates

In the other way around, a decoder $p_i(y'|\hat{x}')$ and a marginal $p_i(\hat{x}')$ determine the subsequent encoder $p_{i+1}(\hat{x}|x)$. Using diagram (A24), one has

$$\begin{aligned} \frac{d \log p_{i+1}(\hat{x}|x)}{d \log p_i(y'|\hat{x}')} &= \frac{\partial \log p_{i+1}(\hat{x}|x)}{\partial \log p_i(y'|\hat{x}')} + \frac{\partial \log p_{i+1}(\hat{x}|x)}{\partial \log Z_i(x_1)} \frac{\partial \log Z_i(x_1)}{\partial \log p_i(y'|\hat{x}')} \\ &= \beta \delta_{\hat{x},x'} p(y'|x) - \delta_{x,x_1} \beta p_{i+1}(\hat{x}'|x_1)p(y'|x_1) \end{aligned} \tag{A67}$$

Summing over the dummy variable x_1 , this is the coordinates' exchange Jacobian J_{dec}^{enc} mentioned in Section 2:

$$\frac{d \log p_{i+1}(\hat{x}|x)}{d \log p_i(y'|\hat{x}')} = \beta p(y'|x) [\delta_{\hat{x},x'} - p_{i+1}(\hat{x}'|x)] \tag{A68}$$

Next, for the derivative with respect to the marginal,

$$\begin{aligned} \frac{d \log p_{i+1}(\hat{x}|x)}{d \log p_i(\hat{x}')} &= \frac{\partial \log p_{i+1}(\hat{x}|x)}{\partial \log p_i(\hat{x}')} + \frac{\partial \log p_{i+1}(\hat{x}|x)}{\partial \log Z_i(x_1)} \frac{\partial \log Z_i(x_1)}{\partial \log p_i(\hat{x}')} \\ &+ \left[\frac{\partial \log p_{i+1}(\hat{x}|x)}{\partial \log Z_i(x_1)} \frac{\partial \log Z_i(x_1)}{\partial \log p_i(y_2|\hat{x}_3)} + \frac{\partial \log p_{i+1}(\hat{x}|x)}{\partial \log p_i(y_2|\hat{x}_3)} \right] \frac{\partial \log p_i(y_2|\hat{x}_3)}{\partial \log p_i(x_4|\hat{x}_5)} \frac{\partial \log p_i(x_4|\hat{x}_5)}{\partial \log p_i(\hat{x}')} \\ &= \delta_{\hat{x},\hat{x}'} - \delta_{x,x_1} p_{i+1}(\hat{x}'|x_1) \\ &+ \left[-\delta_{x,x_1} \beta p_{i+1}(\hat{x}_3|x_1)p(y_2|x_1) + \beta \delta_{\hat{x}_3,\hat{x}} p(y_2|x) \right] \frac{p(y_2|x_4)p_i(x_4|\hat{x}_3)}{p_i(y_2|\hat{x}_3)} \delta_{\hat{x}_3,\hat{x}_5} \cdot (-\delta_{\hat{x}_5,\hat{x}'}) \end{aligned} \tag{A69}$$

Summing over the five dummy variables, this is the coordinates' exchange Jacobian J_{mrg}^{enc} from Section 2:

$$\frac{d \log p_{i+1}(\hat{x}|x)}{d \log p_i(\hat{x}')} = (1 - \beta) [\delta_{\hat{x},\hat{x}'} - p_{i+1}(\hat{x}'|x)] \tag{A70}$$

Finally, note that the encoder Step 1.8 depends on β explicitly, rather than indirectly only via its other variables. So, to calculate the partial derivative term $\frac{\partial \log p_{i+1}(\hat{x}|x)}{\partial \beta}$ in (A63), write as follows for $\log Z$:

$$\begin{aligned} \frac{\partial}{\partial \beta} Z_i(x, \beta) &\stackrel{\text{Step 1.7}}{=} \sum_{\hat{x}} p_i(\hat{x}) \frac{\partial}{\partial \beta} \exp \{ -\beta D_{KL} [p(y|x) || p_i(y|\hat{x})] \} \\ &= - \sum_{\hat{x}} p_i(\hat{x}) D_{KL} [p(y|x) || p_i(y|\hat{x})] \exp \{ -\beta D_{KL} [p(y|x) || p_i(y|\hat{x})] \} \end{aligned} \tag{A71}$$

Thus,

$$\begin{aligned} \frac{\partial}{\partial \beta} \log Z_i(x, \beta) &= \frac{1}{Z_i(x, \beta)} \frac{\partial}{\partial \beta} Z_i(x, \beta) \\ &\stackrel{\text{(A71)}}{=} - \sum_{\hat{x}} \frac{p_i(\hat{x}) \exp \{ -\beta D_{KL} [p(y|x) || p_i(y|\hat{x})] \}}{Z_i(x, \beta)} D_{KL} [p(y|x) || p_i(y|\hat{x})] \\ &\stackrel{\text{Step 1.8}}{=} - \sum_{\hat{x}} p_{i+1}(\hat{x}|x) D_{KL} [p(y|x) || p_i(y|\hat{x})] . \end{aligned} \tag{A72}$$

And so, from the encoder Step 1.8, we have

$$\begin{aligned} \frac{\partial \log p_{i+1}(\hat{x}|x)}{\partial \beta} &= \frac{\partial \log p_i(\hat{x})}{\partial \beta} - \frac{\partial \log Z_i(x, \beta)}{\partial \beta} - \frac{\partial}{\partial \beta} \left(\beta D_{KL}[p(y|x)||p_i(y|\hat{x})] \right) \\ &\stackrel{(A72)}{=} \sum_{\hat{x}''} p_{i+1}(\hat{x}''|x) D_{KL}[p(y|x)||p_i(y|\hat{x}'')] - D_{KL}[p(y|x)||p_i(y|\hat{x})] \end{aligned} \quad (A73)$$

where the term $\frac{\partial \log p_i(\hat{x})}{\partial \beta}$ vanishes since it is considered as an independent variable here.

Appendix C. Proof of Lemma 1, on the Kernel of the Jacobian of the IB Operator in Log-Decoder Coordinates

We prove Lemma 1 from Section 3, using the results of Appendix B.

In the first direction, suppose that $((v_{y,\hat{x}})_{y,\hat{x}'} (u_{\hat{x}})_{\hat{x}})$ is a vector in the left kernel of the Jacobian of the IB operator (5) in log-decoder coordinates, $I - D_{\log p(y|\hat{x}), \log p(\hat{x})} B A_{\beta}$, as in (16) in Section 3. Using the Jacobian’s implicit form (A25) (Appendix B.2), this is to say that

$$v_{y',\hat{x}'} = \sum_{y,\hat{x}} v_{y,\hat{x}} \frac{d \log p_{i+1}(y|\hat{x})}{d \log p_i(y'|\hat{x}')} + \sum_{\hat{x}} u_{\hat{x}} \frac{d \log p_{i+1}(\hat{x})}{d \log p_i(y'|\hat{x}')} \quad \text{and} \quad (A74)$$

$$u_{\hat{x}'} = \sum_{y,\hat{x}} v_{y,\hat{x}} \frac{d \log p_{i+1}(y|\hat{x})}{d \log p_i(\hat{x}')} + \sum_{\hat{x}} u_{\hat{x}} \frac{d \log p_{i+1}(\hat{x})}{d \log p_i(\hat{x}')} \quad (A75)$$

hold, for every y' and \hat{x}' . We spell out and manipulate these equations to obtain the desired result.

By the Jacobian’s explicit form (A44) from Appendix B.2, Equation (A74) spells out as

$$\begin{aligned} v_{y',\hat{x}'} &= \beta \cdot \sum_{y,\hat{x}} v_{y,\hat{x}} \sum_{\hat{x}'',y''} (\delta_{\hat{x}'',\hat{x}'} - \delta_{\hat{x},\hat{x}'}) \cdot \left(1 - \frac{\delta_{y'',y}}{p_{i+1}(y|\hat{x})} \right) C(\hat{x}, \hat{x}''; i+1)_{y',y''} \\ &\quad + \beta \cdot \sum_{\hat{x}} u_{\hat{x}} \left[\delta_{\hat{x},\hat{x}'} p_{i+1}(y'|\hat{x}) - B(\hat{x}, \hat{x}'; i+1)_{y'} \right], \end{aligned} \quad (A76)$$

while the second Equation (A75) spells out as

$$\begin{aligned} u_{\hat{x}'} &= (1 - \beta) \cdot \sum_{y,\hat{x}} v_{y,\hat{x}} \sum_{y''} \left[1 - \frac{\delta_{y'',y}}{p_{i+1}(y|\hat{x})} \right] B(\hat{x}, \hat{x}'; i+1)_{y''} \\ &\quad + (1 - \beta) \cdot \sum_{\hat{x}} u_{\hat{x}} \left(\delta_{\hat{x},\hat{x}'} - A(\hat{x}, \hat{x}'; i+1) \right). \end{aligned} \quad (A77)$$

Next, we expand and simplify each of the terms in (A76) and (A77), using the definition (A32) of A, B , and C from Appendix B.2.

For the first summand to the right of (A76),

$$\begin{aligned} &\beta \cdot \sum_{y,\hat{x}} v_{y,\hat{x}} \sum_{\hat{x}'',y''} (\delta_{\hat{x}'',\hat{x}'} - \delta_{\hat{x},\hat{x}'}) \left(1 - \frac{\delta_{y'',y}}{p_{i+1}(y|\hat{x})} \right) C(\hat{x}, \hat{x}''; i+1)_{y',y''} \\ &\stackrel{(A32)}{=} \beta \cdot \sum_{y,\hat{x}} v_{y,\hat{x}} \sum_{\hat{x}'',y''} (\delta_{\hat{x}'',\hat{x}'} - \delta_{\hat{x},\hat{x}'}) \left(1 - \frac{\delta_{y'',y}}{p_{i+1}(y|\hat{x})} \right) \sum_x p(y'|x) p(y''|x) p_{i+1}(\hat{x}''|x) p_{i+1}(x|\hat{x}) \end{aligned} \quad (A78)$$

We simplify each of the four addends to the right of (A78) while temporarily ignoring the β coefficient. For the $\delta_{\hat{x}'', \hat{x}'} \cdot 1$ term,

$$\begin{aligned} \sum_{y, \hat{x}} v_{y, \hat{x}} \sum_{\hat{x}'', y''} \delta_{\hat{x}'', \hat{x}'} \sum_x p(y'|x) p(y''|x) p_{i+1}(\hat{x}''|x) p_{i+1}(x|\hat{x}) \\ = \sum_{y, \hat{x}} v_{y, \hat{x}} \sum_x p(y'|x) p_{i+1}(\hat{x}'|x) p_{i+1}(x|\hat{x}) \end{aligned} \quad (A79)$$

For the $-\delta_{\hat{x}'', \hat{x}'} \cdot \frac{\delta_{y'', y}}{p_{i+1}(y|\hat{x})}$ term,

$$\begin{aligned} - \sum_{y, \hat{x}} v_{y, \hat{x}} \sum_{\hat{x}'', y''} \delta_{\hat{x}'', \hat{x}'} \delta_{y'', y} \sum_x \frac{1}{p_{i+1}(y|\hat{x})} p(y'|x) p(y''|x) p_{i+1}(\hat{x}''|x) p_{i+1}(x|\hat{x}) \\ = - \sum_{y, \hat{x}} v_{y, \hat{x}} \sum_x \frac{1}{p_{i+1}(y|\hat{x})} p(y'|x) p(y|x) p_{i+1}(\hat{x}'|x) p_{i+1}(x|\hat{x}) \end{aligned} \quad (A80)$$

For the $-\delta_{\hat{x}, \hat{x}'} \cdot 1$ term,

$$\begin{aligned} - \sum_{y, \hat{x}} v_{y, \hat{x}} \sum_{\hat{x}'', y''} \delta_{\hat{x}, \hat{x}'} \sum_x p(y'|x) p(y''|x) p_{i+1}(\hat{x}''|x) p_{i+1}(x|\hat{x}) \\ = - \sum_y v_{y, \hat{x}'} \sum_x p(y'|x) p_{i+1}(x|\hat{x}') = - \sum_y v_{y, \hat{x}'} p_{i+1}(y'|\hat{x}') \end{aligned} \quad (A81)$$

And for the last $-\delta_{\hat{x}, \hat{x}'} \cdot \frac{-\delta_{y'', y}}{p_{i+1}(y|\hat{x})}$ term,

$$\begin{aligned} \sum_{y, \hat{x}} v_{y, \hat{x}} \sum_{\hat{x}'', y''} \delta_{\hat{x}, \hat{x}'} \cdot \frac{\delta_{y'', y}}{p_{i+1}(y|\hat{x})} \sum_x p(y'|x) p(y''|x) p_{i+1}(\hat{x}''|x) p_{i+1}(x|\hat{x}) \\ = \sum_y \frac{v_{y, \hat{x}'}}{p_{i+1}(y|\hat{x}')} \sum_x p(y'|x) p(y|x) p_{i+1}(x|\hat{x}') \end{aligned} \quad (A82)$$

Collecting (A79)–(A82) back into (A78), we obtain

$$\begin{aligned} \beta \cdot \sum_{y, \hat{x}} v_{y, \hat{x}} \sum_x p(y'|x) p_{i+1}(\hat{x}'|x) p_{i+1}(x|\hat{x}) \left[1 - \frac{p(y|x)}{p_{i+1}(y|\hat{x})} \right] \\ + \beta \cdot \sum_y v_{y, \hat{x}'} \frac{1}{p_{i+1}(y|\hat{x}')} \sum_x p(y|x) p(y'|x) p_{i+1}(x|\hat{x}') - \beta \cdot p_{i+1}(y'|\hat{x}') \sum_y v_{y, \hat{x}'} \end{aligned} \quad (A83)$$

for the first summand to the right of (A76).

The second summand to the right of (A76) equals

$$\begin{aligned} \beta \cdot \sum_{\hat{x}} u_{\hat{x}} \left[\delta_{\hat{x}, \hat{x}'} p_{i+1}(y'|\hat{x}) - B(\hat{x}, \hat{x}'; i+1)_{y'} \right] \\ \stackrel{(A32)}{=} \beta \cdot u_{\hat{x}'} p_{i+1}(y'|\hat{x}') - \beta \cdot \sum_x p(y'|x) p_{i+1}(\hat{x}'|x) \sum_{\hat{x}} u_{\hat{x}} p_{i+1}(x|\hat{x}) \end{aligned} \quad (A84)$$

Combining (A83) and (A84), Equation (A76) is equivalent to

$$\begin{aligned} \frac{1}{\beta} \cdot v_{y', \hat{x}'} + p_{i+1}(y'|\hat{x}') \sum_y v_{y, \hat{x}'} - u_{\hat{x}'} p_{i+1}(y'|\hat{x}') \\ = \sum_{y, \hat{x}} v_{y, \hat{x}} \sum_x p(y'|x) p_{i+1}(\hat{x}'|x) p_{i+1}(x|\hat{x}) \left[1 - \frac{p(y|x)}{p_{i+1}(y|\hat{x})} \right] \\ + \sum_y v_{y, \hat{x}'} \sum_x p(y'|x) p_{i+1}(x|\hat{x}') \frac{p(y|x)}{p_{i+1}(y|\hat{x}')} - \sum_x p(y'|x) p_{i+1}(\hat{x}'|x) \sum_{\hat{x}} u_{\hat{x}} p_{i+1}(x|\hat{x}) \end{aligned} \quad (A85)$$

for any y' and \hat{x}' . Summing (A85) over y' and simplifying, we obtain

$$\begin{aligned} & \frac{1}{\beta} \cdot \sum_y v_{y,\hat{x}'} - u_{\hat{x}'} \\ &= \sum_{y,\hat{x}} v_{y,\hat{x}} \sum_x p_{i+1}(\hat{x}'|x) p_{i+1}(x|\hat{x}) \left[1 - \frac{p(y|x)}{p_{i+1}(y|\hat{x})} \right] - \sum_x p_{i+1}(\hat{x}'|x) \sum_{\hat{x}} u_{\hat{x}} p_{i+1}(x|\hat{x}) \end{aligned} \quad (A86)$$

for any \hat{x}' .

Next, we expand and simplify Equation (A77). Using the definition (A32) of B , the first summand to its right can be written as

$$(1 - \beta) \cdot \sum_{y,\hat{x}} v_{y,\hat{x}} \sum_x p_{i+1}(\hat{x}'|x) p_{i+1}(x|\hat{x}) \left[1 - \frac{p(y|x)}{p_{i+1}(y|\hat{x})} \right]. \quad (A87)$$

Similarly, the second summand to the right of (A77) can be written as

$$(1 - \beta) \cdot \left[u_{\hat{x}'} - \sum_x p_{i+1}(\hat{x}'|x) \sum_{\hat{x}} u_{\hat{x}} p_{i+1}(x|\hat{x}) \right]. \quad (A88)$$

Combining (A87) and (A88), Equation (A77) can now be written explicitly,

$$\frac{\beta}{1-\beta} \cdot u_{\hat{x}'} = \sum_{y,\hat{x}} v_{y,\hat{x}} \sum_x p_{i+1}(\hat{x}'|x) p_{i+1}(x|\hat{x}) \left[1 - \frac{p(y|x)}{p_{i+1}(y|\hat{x})} \right] - \sum_x p_{i+1}(\hat{x}'|x) \sum_{\hat{x}} u_{\hat{x}} p_{i+1}(x|\hat{x}) \quad (A89)$$

for every \hat{x}' . Next, subtracting (A89) from (A86), we obtain

$$u_{\hat{x}} = \frac{1-\beta}{\beta} \cdot \sum_{y,\hat{x}} v_{y,\hat{x}} \quad (A90)$$

for any \hat{x} .

Substituting (A90) into (A85) and using the decoder Step 1.6 to expand $p_{i+1}(y'|\hat{x}')$ there,

$$\begin{aligned} \frac{1}{\beta} \cdot v_{y',\hat{x}'} &= \sum_{y,\hat{x}} v_{y,\hat{x}} \sum_x p_{i+1}(\hat{x}'|x) p(y'|x) p_{i+1}(x|\hat{x}) \left[\frac{2\beta-1}{\beta} - \frac{p(y|x)}{p_{i+1}(y|\hat{x})} \right] \\ &\quad - \sum_y v_{y,\hat{x}'} \sum_x p(y'|x) p_{i+1}(x|\hat{x}') \cdot \frac{2\beta-1}{\beta} + \sum_y v_{y,\hat{x}'} \sum_x p(y'|x) p_{i+1}(x|\hat{x}') \frac{p(y|x)}{p_{i+1}(y|\hat{x}')} \end{aligned} \quad (A91)$$

Next, inserting $\sum_{\hat{x}} \delta_{\hat{x},\hat{x}'}$ into the sums on the last line,

$$\begin{aligned} \frac{1}{\beta} \cdot v_{y',\hat{x}'} &= \sum_{y,\hat{x}} v_{y,\hat{x}} \sum_x p_{i+1}(\hat{x}'|x) p(y'|x) p_{i+1}(x|\hat{x}) \left[\frac{2\beta-1}{\beta} - \frac{p(y|x)}{p_{i+1}(y|\hat{x})} \right] \\ &\quad - \sum_{y,\hat{x}} v_{y,\hat{x}} \sum_x \delta_{\hat{x},\hat{x}'} p(y'|x) p_{i+1}(x|\hat{x}) \left[\frac{2\beta-1}{\beta} - \frac{p(y|x)}{p_{i+1}(y|\hat{x})} \right] \end{aligned} \quad (A92)$$

Finally, this simplifies to

$$v_{y',\hat{x}'} = \sum_{y,\hat{x}} v_{y,\hat{x}} \sum_x p(y'|x) \left[\delta_{\hat{x},\hat{x}'} - p_{i+1}(\hat{x}'|x) \right] p_{i+1}(x|\hat{x}) \left[\beta \cdot \frac{p(y|x)}{p_{i+1}(y|\hat{x})} + (1 - 2\beta) \right] \quad (A93)$$

The latter is to say that $(v_{y,\hat{x}})_{y,\hat{x}}$ is a left-eigenvector of the eigenvalue 1 of the matrix to the right. At an IB root, this is precisely the matrix S (17) from the Lemma's statement, as desired.

As a side note, we comment that Equations (A74) and (A75) also imply

$$\forall y \sum_{\hat{x}} v_{y,\hat{x}} = 0 \quad \text{and} \quad \sum_{\hat{x}} u_{\hat{x}} = 0, \quad (A94)$$

which can be seen by summing (A85) and (A89), respectively, over \hat{x}' , and simplifying.

In the other direction, let $v := (v_{y,\hat{x}})_{y,\hat{x}}$ be a left-eigenvector of the eigenvalue 1 of S (17). That is, assume that Equation (A93) holds. Define a vector $u := (u_{\hat{x}})_{\hat{x}}$ by Equation (A90). Reversing the algebra, (A93) is equivalent to (A91). Substituting (A90) into the latter yields back (A85), which is equivalent to the explicit form (A76) of Equation (A74). Next, summing (A85) over y' and simplifying yields (A86). Adding the latter to (A90) yields back (A89), which is equivalent to Equation (A77), the explicit form of (A75). To conclude, both of the Equations (A74) and (A75) hold, as claimed.

Appendix D. Approximate Error Analysis for Deterministic Annealing and for Euler’s Method with BA

Complementing the results of Section 4, we provide an approximate error analysis for two computation methods for the IB: deterministic annealing and Euler’s method combined with a fixed number of BA iterations.

First, we recap the linearization argument around Equation (10) in [4]. Denote repeated BA iterations initialized at p_0 by

$$p_{k+1} := BA_{\beta}[p_k]. \tag{A95}$$

Linearizing around a fixed-point p_{β} of BA,

$$BA[p_k] \simeq p_{\beta} + D BA_{\beta}|_{p_{\beta}} \cdot (p_k - p_{\beta}), \tag{A96}$$

where $D BA_{\beta}|_{p_{\beta}}$ denotes the Jacobian matrix of BA_{β} evaluated at p_{β} . Rewriting in terms of the error $\delta p_k := p_k - p_{\beta}$ of the k -th iterate,

$$\delta p_{k+1} \simeq D BA_{\beta}|_{p_{\beta}} \cdot \delta p_k. \tag{A97}$$

Thus, to first order, repeated applications of BA_{β} reduce the initial error according to

$$\|\delta p_{k+1}\| \simeq \left\| \left(D BA_{\beta}|_{p_{\beta}} \right)^k \cdot \delta p_0 \right\|. \tag{A98}$$

Next, consider $k > 0$ applications of $BA_{\beta+\Delta\beta}$ to a root p_{β} at β . This is similar to deterministic annealing, but with a capped number of BA iterations. Plugging the initial error $\delta p_0 := p_{\beta} - p_{\beta+\Delta\beta} \simeq -\Delta\beta \frac{dp}{d\beta}|_{\beta}$ into Equation (A98) shows that this method is of the first order:

$$\|\delta p_{k+1}\| \simeq |\Delta\beta| \cdot \left\| \left(D BA_{\beta+\Delta\beta}|_{p_{\beta+\Delta\beta}} \right)^k \frac{dp}{d\beta}|_{\beta} \right\|. \tag{A99}$$

Finally, we combine BA with Euler’s method for the IB, Equation (22). Consider $k > 0$ applications of $BA_{\beta+\Delta\beta}$ to the approximation $p_{\beta} + \Delta\beta \frac{dp}{d\beta}|_{\beta}$ produced by an Euler method step. Its initial error is

$$\delta p_0 := p_{\beta} + \Delta\beta \frac{dp}{d\beta}|_{\beta} - p_{\beta+\Delta\beta} = -\frac{1}{2}(\Delta\beta)^2 \frac{d^2p}{d\beta^2}|_{\beta'}, \tag{A100}$$

where the last equality follows from the second-order expansion $p_{\beta+\Delta\beta} = p_{\beta} + \Delta\beta \frac{dp}{d\beta}|_{\beta} + \frac{1}{2}(\Delta\beta)^2 \frac{d^2p}{d\beta^2}|_{\beta'}$, with $\beta' \in [\beta, \beta + \Delta\beta]$. Similar to before, plugging this into Equation (A98) shows that this method is of the second order:

$$\|\delta p_{k+1}\| \simeq \frac{1}{2}|\Delta\beta|^2 \cdot \left\| \left(D BA_{\beta+\Delta\beta}|_{p_{\beta+\Delta\beta}} \right)^k \frac{d^2p}{d\beta^2}|_{\beta'} \right\|. \tag{A101}$$

Appendix E. An Exact Solution for a Binary Symmetric Channel

Define an IB problem by $Y \sim \text{Bernoulli}(\frac{1}{2})$ and $X := Y \oplus Z$ for $Z \sim \text{Bernoulli}(\alpha)$ independent of Y , $0 < \alpha < \frac{1}{2}$, where \oplus denotes addition modulo 2. Explicitly, it is given by $p_{Y|X} = \begin{pmatrix} 1-\alpha & \alpha \\ \alpha & 1-\alpha \end{pmatrix}$ and $p_X = (\frac{1}{2}, \frac{1}{2})$. We synthesize exact solutions for this problem using Mrs. Gerber’s Lemma [38] and by following [2].

Let $h(p) := -p \log p - (1-p) \log(1-p)$ be the binary entropy, with $h(0) := h(1) := 0$. It is injective on $[0, \frac{1}{2}]$, with a maximal value of $\log 2$ at $p = \frac{1}{2}$. So, its inverse function h^{-1} is well-defined on $[0, \log 2]$. Given a constraint $I_X \in [0, \log 2]$ on $I(\hat{X}; X)$, $I(\hat{X}; X) \leq I_X$, define a random variable $V \sim \text{Bernoulli}(\delta)$ and set $\hat{X} := X \oplus V$, where δ is defined by $h(\delta) = \log 2 - I_X$ or equivalently in terms of h^{-1} by $\delta := h^{-1}(\log 2 - I_X)$. Explicitly, $p(\hat{x}|x) = \begin{pmatrix} 1-\delta & \delta \\ \delta & 1-\delta \end{pmatrix}$, with its rows indexed by \hat{x} and columns by x . \hat{X} is also a $\text{Bernoulli}(\frac{1}{2})$ variable since X is, and so

$$I(\hat{X}; X) = H(\hat{X}) - H(\hat{X}|X) = \log 2 - h(\delta) = I_X, \tag{A102}$$

showing that the constraint on $I(\hat{X}; X)$ holds. The chain $\hat{X} \rightarrow X \rightarrow Y$ of random variables is readily seen to be Markov. By Corollary 4 in [38], it follows that $I(\hat{X}; Y) \leq \log 2 - h(\alpha * \delta)$, where $a * b := a(1-b) + b(1-a)$. Finally, equality follows by Theorem 1 there. Thus, the above $p(\hat{x}|x)$ is IB-optimal.

The above defines an IB solution $p(\hat{x}|x)$ as a function of I_X . However, our numerical computations are phrased in terms of the IB’s Lagrange multiplier β . To that end, note that [2] (Section IV.A) show that

$$\beta \cdot (1 - 2\alpha) \log \frac{1 - \alpha * \delta}{\alpha * \delta} = \log \frac{1 - \delta}{\delta}, \tag{A103}$$

and that the bifurcation of this problem occurs at

$$\beta_c = \frac{1}{(1 - 2\alpha)^2}. \tag{A104}$$

To conclude, we have $\beta = \beta(\delta)$ as a function of δ , $\delta = \delta(I_X)$ as a function of I_X , and the encoder $p(\hat{x}|x)$ as a function of δ . These functional dependencies are summarized as follows:



where the variable at the tail of each arrow is a function of that at its head.

Writing $\mathbf{p} = (p(\hat{x}|x))_{\hat{x},x}$, its derivative with respect to β can be calculated by the chain rule:

$$\frac{d\mathbf{p}}{d\beta} = \frac{d}{d\beta} \left(\mathbf{p}(\beta^{-1}(\delta)) \right) = \frac{d\mathbf{p}}{d\delta} \left(\frac{d\beta}{d\delta} \right)^{-1}, \tag{A106}$$

where we have applied the derivative of an inverse function $(f^{-1})' = 1/f'$ to $\beta(\delta)$ in (A103), to differentiate $\delta(\beta)$. From the argument around (A102), $\frac{d\mathbf{p}}{d\delta} = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$. While this yields an analytical expression for the derivative $\frac{d\mathbf{p}}{d\beta}$, both of the terms to the right of (A106) are evaluated at $\delta(\beta)$, for a given β value. Although it is straightforward to compute $\delta(\beta)$ numerically from (A103), this entails numerical error, especially as δ approaches $1/2$ near the bifurcation. For the solution with respect to decoder coordinates, an immediate application of the Bayes rule shows that

$$p(\hat{x}) = \frac{1}{2} \quad \text{and} \quad p(y|\hat{x}) = \begin{pmatrix} \alpha * (1-\delta) & \alpha * \delta \\ \alpha * \delta & \alpha * (1-\delta) \end{pmatrix}, \tag{A107}$$

where the rows of $p(y|\hat{x})$ are indexed by y , and columns by \hat{x} . Along with $\frac{dp(y|\hat{x})}{d\beta} = (2\alpha - 1) \cdot \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$, its derivatives with respect to β follow as in (A106).

Appendix F. Equivalent Conditions for Cluster-Merging Bifurcations

We briefly discuss the equivalent conditions for cluster-merging bifurcations in the IB (Section 5.2) found in the literature.

Rose et al. [39] (Section 4) derive a condition for cluster-splitting phase transitions (Equation (17) there) in the context of fuzzy clustering. Following this, [13] (3.2 in Part III) derives an analogous condition for cluster splitting in the IB (Equation (12) there):

$$(I - \beta C_X(\hat{x}; \beta))\mathbf{u} = \mathbf{0}, \tag{A108}$$

where I is the identity. Namely, for a cluster \hat{x} to split, it is necessary that $1/\beta$ be an eigenvalue of an $|\mathcal{X}'|$ -by- $|\mathcal{X}'|$ matrix $C_X(\hat{x}; \beta)$, whose entries at an IB root are given by

$$C_X(\hat{x}; \beta)_{x,x'} := \sum_y \frac{p(y|x)p(y|x')p_\beta(x'|\hat{x})}{p_\beta(y|\hat{x})}. \tag{A109}$$

While the coefficients matrix (A109) for the IB differs from the one for fuzzy clustering, inter-cluster interactions are explicitly neglected in both derivations (see therein). Indeed, the definition (A109) of C_X involves the coordinates of cluster \hat{x} alone, as one might expect when considering a root in either decoder or in inverse encoder coordinates (Section 2). Reversing the dynamics in β , condition (A108) characterizes cluster-merging bifurcations in the IB (Section 5.2).

In [13] it is noted that (A108) is closely related to the bifurcation analysis of [9]. The latter provides a condition to identify the critical β values of IB bifurcations, given in their Theorem 5.3. Indeed, their condition is equivalent to (A108), and therefore it also characterizes cluster-merging bifurcations. To see this, the necessary condition they give for a phase transition at β is that $1/\beta$ must be an eigenvalue of a matrix V (Equation (21) there). When written in our notation, this matrix is given by

$$V(\hat{x}; \beta)_{x,x'} := \sum_y \frac{p(x',y)p(x,y)p_\beta(\hat{x}|x)}{p_\beta(y,\hat{x})p(x')}. \tag{A110}$$

However, V (A110) is readily seen to be the transpose of C_X (A109), and so they have the same eigenvalues.

Appendix G. Lyapunov Stability of an Optimal IB Root

We provide the essential parts of a proof that an optimal IB root is Lyapunov uniformly asymptotically stable on closed intervals which do not contain a bifurcation when following the flow dictated by the IB's ODE (16) in *decreasing* β . Definitions for the below are as in [40] (see especially Section 4.2 there). See Section 6.3 for a discussion of the results below.

Let $\mathbf{p}^*(\beta)$ be an optimal IB root. We start by rewriting it as an equilibrium of a non-autonomous ODE, as in [40] (Equation (4.1)). Consider the implicit ODE (7) $\frac{d\mathbf{p}}{d\beta} = -(D_{\mathbf{p}}F)^{-1}D_{\beta}F$, specialized to the IB by setting $F := Id - BA_{\beta}$ (5). Denote $\delta\mathbf{p} := \mathbf{p} - \mathbf{p}^*$, for an arbitrary \mathbf{p} . Subtracting the ODE at \mathbf{p} from that at \mathbf{p}^* yields a non-autonomous ODE in the error $\delta\mathbf{p}$ from the optimal root:

$$\frac{d\delta\mathbf{p}}{d\beta} = (D_{\mathbf{p}}F)^{-1}D_{\beta}F|_{\mathbf{p}^*} - (D_{\mathbf{p}}F)^{-1}D_{\beta}F|_{\mathbf{p}^*+\delta\mathbf{p}} \tag{A111}$$

This rewrites the given root \mathbf{p}^* as an equilibrium $\delta\mathbf{p} = \mathbf{0}$ of this ODE (A111), simplifying the below.

Next, we define a Lyapunov function for the flow of the equilibrium $\delta p = 0$ along the ODE (A111), when its dynamics in β is reversed. Consider the IB's Lagrangian $\mathcal{L}_\beta := I(X; \hat{X}) - \beta \cdot I(Y; \hat{X})$ as a functional in p , and let $\mathcal{L}_\beta^* := \mathcal{L}_\beta[p^*]$ be its optimal value at β . Then,

$$(\mathcal{L}_\beta - \mathcal{L}_\beta^*)(\delta p) \tag{A112}$$

is the desired Lyapunov function. Specifically, (i) $\mathcal{L}_\beta - \mathcal{L}_\beta^*$ is positive definite and (ii) $\frac{d}{d\beta}(\mathcal{L}_\beta - \mathcal{L}_\beta^*)$ is negative definite, when the dynamics in β are reversed. Theorem 4.1 in [40] then implies that $\delta p = 0$ is uniformly asymptotically stable [40] (Definition 4.6).

For (i), $\mathcal{L}_\beta - \mathcal{L}_\beta^*$ (A112) is immediately seen to be positive semi-definite from the definition of \mathcal{L}_β^* , up to technicalities ignored here (cf., Definition 4.7 in [40]). The results of Section 5.3 (after Proposition 1) imply that representing p in reduced log-decoder coordinates renders (A112) strictly positive definite. Indeed, $D(Id - BA_\beta)$ is non-singular in a reduced representation in these coordinates, as mentioned there, and so an optimal root p^* is locally unique. As for condition (ii), from the definition of \mathcal{L}_β we have

$$\frac{d}{d\beta} \mathcal{L}_\beta = \frac{d}{d\beta} I(X; \hat{X}) - \beta \frac{d}{d\beta} I(Y; \hat{X}) - I(Y; \hat{X}) = -I(Y; \hat{X}), \tag{A113}$$

where $\frac{d}{d\beta} I(X; \hat{X}) = \beta \frac{d}{d\beta} I(Y; \hat{X})$ in the last equality follows by direct calculations similar to those in the Appendix of [13] (Part III). Thus, for the β -derivative of (A112), we have

$$\frac{d}{d\beta} (\mathcal{L}_\beta - \mathcal{L}_\beta^*)(\delta p) = I(Y; \hat{X})|_{p^*} - I(Y; \hat{X})|_p. \tag{A114}$$

The latter is always positive semi-definite around p^* , since by definition (1) p^* yields the maximal Y -information subject to a constraint on the X -information. The same argument as above shows that it is strictly positive definite. Finally, reversing the dynamics in β leaves the ODE (A111) unaffected but flips the sign of (A114), rendering it negative definite as required.

Appendix H. Introducing Degeneracies to the IB Operator in Decoder Coordinates Is Uninformative at an Isolated Optimal Root

Suppose that the Jacobian of the IB operator $Id - BA_\beta$ (5) in log-decoder coordinates is non-singular at a reduced representation of an IB root, as in the argument following Proposition 1 (Section 5.3). We show that evaluating it on a non-reduced (degenerate) representation only permits one to detect kernel directions which are due to the selected degeneracy. Thus, it is inadequate for detecting bifurcations, as explained in Section 5.3.

Let $p \in \Delta[\Delta[\mathcal{Y}]]$ be an IB root of effective cardinality T_1 . A T -clustered representation of a root (e.g., in decoder coordinates) is a function $\pi : \Delta[\Delta[\mathcal{Y}]] \rightarrow \mathbb{R}^{(|\mathcal{Y}|+1) \cdot T}$, defined on some neighborhood of p . In the other way around, one can consider the inclusion $i : \mathbb{R}^{(|\mathcal{Y}|+1) \cdot T} \rightarrow \Delta[\Delta[\mathcal{Y}]]$, defined in the obvious way on normalized decoder coordinates at some neighborhood of $\pi(p)$. Let π be a representation of p in its effective cardinality T_1 , and $\tilde{\pi}$ a degenerate one on $T_2 > T_1$ clusters. These satisfy

$$\pi = \text{reduc} \circ \tilde{\pi}, \tag{A115}$$

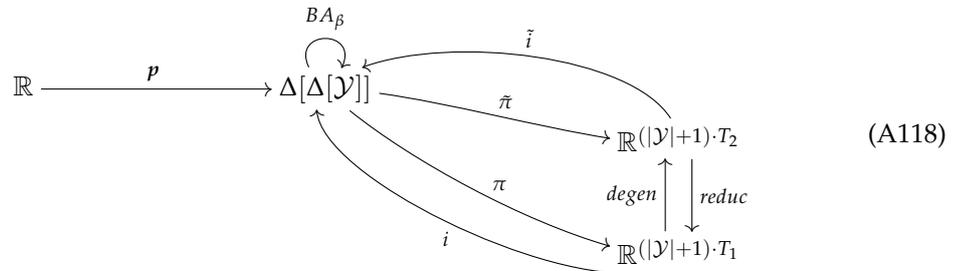
where *reduc* is the reduction map (defined similar to the root-reduction Algorithm 2, by setting its thresholds to zero, $\delta_1 = \delta_2 = 0$, and replacing its strict inequalities with non-strict ones. Note that Algorithm 2 has a well-defined output for every input). In the other way around, one can pick a particular degenerating map *degen* (e.g., "split the third cluster to two copies of probability ratio 1:2"). Applying a particular degeneracy and then reducing is the identity,

$$\text{reduc} \circ \text{degen} = Id, \tag{A116}$$

though not the other way around. Let i and \tilde{i} be the inclusions corresponding to π and $\tilde{\pi}$, respectively. Similar to (A115), degenerating a root has no effect before it is included into $\Delta[\Delta[\mathcal{Y}]]$,

$$i = \tilde{i} \circ \text{degen} \tag{A117}$$

Recall from Section 5.1 (before Conjecture 1) that BA_β in decoder coordinates may be considered as an operator on $\Delta[\Delta[\mathcal{Y}]]$. To summarize, we have the following diagram:



Next, consider the representations of the IB operator $Id - BA_\beta$ (5) on T_1 and T_2 clusters. These amount to pre-composing with the inclusions and post-composing with the representation maps. Denote by Id_i the identity operator on $\mathbb{R}^{(|\mathcal{Y}|+1) \cdot T_i}$. By identities (A115), (A116), and (A117), the T_1 -clustered representation of $Id - BA_\beta$ (5) satisfies

$$\begin{aligned} Id_1 - \pi \circ BA_\beta \circ i &= \text{reduc} \circ \text{degen} - \text{reduc} \circ \tilde{\pi} \circ BA_\beta \circ \tilde{i} \circ \text{degen} \\ &= \text{reduc} \circ [Id_2 - \tilde{\pi} \circ BA_\beta \circ \tilde{i}] \circ \text{degen} \end{aligned} \tag{A119}$$

Differentiating, by the chain rule we have

$$D(Id_1 - \pi \circ BA_\beta \circ i) = D(\text{reduc})D(Id_2 - \tilde{\pi} \circ BA_\beta \circ \tilde{i})D(\text{degen}). \tag{A120}$$

Multiplying a matrix B from the left can only enlarge the kernel, $\dim \ker(AB) \geq \dim \ker B$, and so

$$\dim \ker D(Id_1 - \pi \circ BA_\beta \circ i) \geq \dim \ker (D(Id_2 - \tilde{\pi} \circ BA_\beta \circ \tilde{i})D(\text{degen})). \tag{A121}$$

Since the left-hand side is evaluated at a reduced representation, it vanishes by assumption. Thus, $D(Id_2 - \tilde{\pi} \circ BA_\beta \circ \tilde{i})D(\text{degen})$ is of full rank, yielding

$$\ker D(Id_2 - \tilde{\pi} \circ BA_\beta \circ \tilde{i}) \subset (\text{Im } D(\text{degen}))^\perp, \tag{A122}$$

where $(\text{Im } D(\text{degen}))^\perp$ denotes the vectors tangent to the column space $\text{Im } D(\text{degen})$ of $D(\text{degen})$. Stated differently, the Jacobian $D(Id_2 - \tilde{\pi} \circ BA_\beta \circ \tilde{i})$ of a T_2 -clustered representation of $Id - BA_\beta$ (5) can only detect directions in $(\text{Im } D(\text{degen}))^\perp$, which are determined by the choice of the degenerating map degen , as argued.

For completeness, splitting a cluster $r \in \Delta[\mathcal{Y}]$ into two at some fixed ratio $0 < \lambda < 1$ is of the form $(r, p(r)) \mapsto (r, \lambda \cdot p(r), r, (1 - \lambda) \cdot p(r))$. Adding a pre-defined cluster $r' \in \Delta[\mathcal{Y}]$ of zero mass is constant $(\dots) \mapsto (\dots, r', 0)$ on the newly added coordinates. A general degeneracy map degen is a composition of these, and is otherwise the identity map on unaffected coordinates.

References

1. Tishby, N.; Pereira, F.C.; Bialek, W. The Information Bottleneck Method. In Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, 22–24 September 1999; pp. 368–377.
2. Witsenhausen, H.; Wyner, A. A Conditional Entropy Bound for a Pair of Discrete Random Variables. *IEEE Trans. Inf. Theory* **1975**, *21*, 493–501. [CrossRef]

3. Zaidi, A.; Estella-Aguerri, I.; Shamai, S. On the Information Bottleneck Problems: Models, connections, Applications and Information Theoretic Views. *Entropy* **2020**, *22*, 151. [CrossRef] [PubMed]
4. Agmon, S.; Bengier, E.; Ordentlich, O.; Tishby, N. Critical Slowing Down Near Topological Transitions in Rate-Distortion Problems. In Proceedings of the 2021 IEEE International Symposium on Information Theory (ISIT), Melbourne, Australia, 12–20 July 2021; pp. 2625–2630. [CrossRef]
5. Gilad-Bachrach, R.; Navot, A.; Tishby, N. An Information Theoretic Tradeoff between Complexity and Accuracy. In *Learning Theory and Kernel Machines*; Springer: Berlin/Heidelberg, Germany, 2003. [CrossRef]
6. Agmon, S. Root Tracking for Rate-Distortion: Approximating a Solution Curve with Higher Implicit Multivariate Derivatives. *IEEE Trans. Inf. Theory* **2023**, *in press*.
7. De Oliveira, O. The Implicit and the Inverse Function Theorems: Easy Proofs. *Real Anal. Exch.* **2014**, *39*, 207–218. [CrossRef]
8. Blahut, R. Computation of Channel Capacity and Rate-Distortion Functions. *IEEE Trans. Inf. Theory* **1972**, *18*, 460–473. [CrossRef]
9. Gedeon, T.; Parker, A.E.; Dimitrov, A.G. The Mathematical Structure of Information Bottleneck Methods. *Entropy* **2012**, *14*, 456–479. [CrossRef]
10. Agmon, S. On Bifurcations in Rate-Distortion Theory and the Information Bottleneck Method. Ph.D. Thesis, The Hebrew University of Jerusalem, Jerusalem, Israel, 2022.
11. Rose, K.; Gurewitz, E.; Fox, G. A deterministic annealing approach to clustering. *Pattern Recognit. Lett.* **1990**, *11*, 589–594. [CrossRef]
12. Kuznetsov, Y.A. *Elements of Applied Bifurcation Theory*, 3rd ed.; Springer Science & Business Media: New York, NY USA, 2004; Volume 112. [CrossRef]
13. Zaslavsky, N. Information-Theoretic Principles in the Evolution of Semantic Systems. Ph.D. Thesis, The Hebrew University of Jerusalem, Jerusalem, Israel, 2019.
14. Ngampruetikorn, V.; Schwab, D.J. Perturbation Theory for the Information Bottleneck. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 21008–21018. [PubMed]
15. Wu, T.; Fischer, I.; Chuang, I.L.; Tegmark, M. Learnability for the Information Bottleneck. *PMLR* **2020**, *115*, 1050–1060. [CrossRef]
16. Wu, T.; Fischer, I. Phase Transitions for the Information Bottleneck in Representation Learning. In Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020), Virtual Conference, 26 April–1 May 2020.
17. Rose, K. A Mapping Approach to Rate-Distortion Computation and Analysis. *IEEE Trans. Inf. Theory* **1994**, *40*, 1939–1952. [CrossRef]
18. Giaquinta, M.; Hildebrandt, S. *Calculus of Variations I*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 310. [CrossRef]
19. Parker, A.E.; Dimitrov, A.G. Symmetry-Breaking Bifurcations of the Information Bottleneck and Related Problems. *Entropy* **2022**, *24*, 1231. [CrossRef] [PubMed]
20. Harremoës, P.; Tishby, N. The Information Bottleneck Revisited or How to Choose a Good Distortion Measure. In Proceedings of the 2007 IEEE International Symposium on Information Theory, Nice, France, 24–29 June 2007; pp. 566–570. [CrossRef]
21. Kielhöfer, H. *Bifurcation Theory: An Introduction with Applications to Partial Differential Equations*, 2nd ed.; Springer: New York, NY, USA, 2012. [CrossRef]
22. Lee, J.M. *Introduction to Smooth Manifolds*, 2nd ed.; Springer: New York, NY, USA, 2012. [CrossRef]
23. Dummit, D.S.; Foote, R.M. *Abstract Algebra*, 3rd ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2004.
24. Teschl, G. *Topics in Linear and Nonlinear Functional Analysis*; University of Vienna: Vienna, Austria, 2022. Available online: <https://www.mat.univie.ac.at/~gerald/ftp/book-fa/fa.pdf> (accessed on 20 December 2022).
25. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2001.
26. Butcher, J.C. *Numerical Methods for Ordinary Differential Equations*, 3rd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2016. [CrossRef]
27. Atkinson, K.E.; Han, W.; Stewart, D. *Numerical Solution of Ordinary Differential Equations*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 108. [CrossRef]
28. Berger, T. *Rate Distortion Theory: A Mathematical Basis for Data Compression*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1971.
29. Shannon, C.E. A Mathematical Theory of Communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
30. Shannon, C.E. Coding Theorems for a Discrete Source with a Fidelity Criterion. *IRE Nat. Conv. Rec.* **1959**, *4*, 325–350.
31. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*, 2nd ed.; John Wiley & Sons: Hoboken, NJ, USA, 2006; p. 748. [CrossRef]
32. Dieudonné, J. *Foundations of Modern Analysis*; Academic Press: Cambridge, MA, USA, 1969.
33. Gowers, T.; Barrow-Green, J.; Leader, I. *The Princeton Companion to Mathematics*; Princeton University Press: Princeton, NJ, USA, 2008.
34. Coolidge, J.L. *A Treatise on Algebraic Plane Curves*; Dover: Dover, NY, USA, 1959.
35. Strogatz, S.H. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2018.
36. Golubitsky, M.; Stewart, I.; Schaeffer, D.G. *Singularities and Groups in Bifurcation Theory II*; Springer: New York, NY, USA, 1988. [CrossRef]
37. Bengier, E. (The Hebrew University of Jerusalem, Jerusalem, Israel). Private communications, 2019.
38. Wyner, A.; Ziv, J. A Theorem on the Entropy of Certain Binary Sequences and Applications: Part I. *IEEE Trans. Inf. Theory* **1973**, *19*, 769–772. [CrossRef]

39. Rose, K.; Gurewitz, E.; Fox, G.C. Statistical Mechanics and Phase Transitions in Clustering. *Phys. Rev. Lett.* **1990**, *65*, 945. [[CrossRef](#)] [[PubMed](#)]
40. Slotine, J.J.E.; Li, W. *Applied Nonlinear Control*; Prentice Hall: Englewood Cliffs, NJ, USA, 1991; Volume 199.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.