SPARLING: LEARNING LATENT REPRESENTATIONS WITH EXTREMELY SPARSE ACTIVATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world processes often contain intermediate state that can be modeled as an extremely sparse activation tensor. In this work, we analyze the identifiability of such sparse and local latent intermediate variables, which we call *motifs*. We prove our Motif Identifiability Theorem, stating that under certain assumptions it is possible to precisely identify these motifs exclusively by reducing end-to-end error. Additionally, we provide the SPARLING algorithm, which uses a new kind of informational bottleneck that enforces levels of activation sparsity unachievable using other techniques. We find that extreme sparsity is necessary to achieve good intermediate state modeling empirically. On our synthetic DIGITCIRCLE domain as well as the LATEX-OCR and AUDIOMNISTSEQUENCE domains, we are able to precisely localize the intermediate states up to feature permutation with $> 90\%$ accuracy, even though we only train end-to-end.

1 INTRODUCTION

A hallmark of deep learning is its ability to learn useful intermediate representations of data from end-to-end supervision via backpropagation. However, these representations are often opaque: values in the intermediate vectors do not always map to semantically meaningful concepts. Concept bottlenecks (Koh et al. (2020)) have been proposed as a solution to this problem, labels for meaningful intermediate concepts help align signals in the intermediate layers with these concepts.

Recent work in Genomics has demonstrated both the desirability and the feasibility of learning meaningful intermediate representations in the context of RNA splicing, a task where a function is learned to map RNA to a set of annotations identifying the boundary points between coding and non-coding regions. In Gupta et al. (2024), the authors attempt to identify the mechanism of splicing by breaking the end-to-end function into a motif identification function that maps the input RNA sequence to a set of annotations representing the binding sites of proteins (a "motif" being a nonzero entry in this intermediate activation tensor), and an aggregator function that maps these binding sites to the output boundary annotations. The paper showed that an approach similar to concept bottlenecks could reconstruct the desired mechanism when supplemented with an extreme sparsity constraint reflecting the extreme sparsity of the binding sites in the true biological mechanism.

This raises an important theoretical question regarding the extent to which extreme sparsity alone, without additional intermediate annotations, can be used to identify an intermediate latent variable. In this work, we prove that under certain circumstances, extreme sparsity and end-to-end training alone are sufficient to identify an intermediate latent variable. Additionally, we provide and demonstrate an algorithm that is capable of such identification on three different domains.

Example. Figure 1 shows a sample task we call DIGITCIRCLE and compares it to the splicing task. The input is a noisy image of a circle of digits, and the output is a list of the digits read counterclockwise starting from the smallest. The key question for this paper is whether it is possible to learn to recognize individual digits given only end-to-end data (paired examples of x and y^*).

Contributions. We present three main contributions in this paper. First, we provide a proof of our Motif Identifiability Theorem: that sparse local latent variables are identifiable. We attempt to make as few assumptions as possible about the structure of the relationships between the inputs, motifs, and output, assuming only that the motif patches are separated and independent from each other and are relevant to computing the output. We do not make any further assumptions regarding

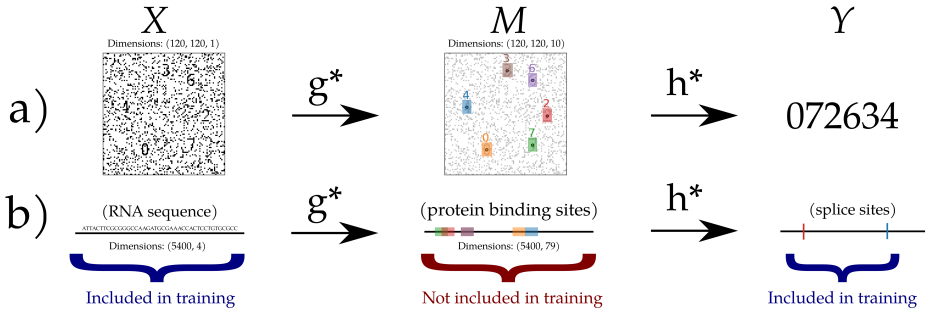

Figure 1: (a) Example of the DIGITCIRCLE domain, alongside (b) a cartoon of the splicing problem. The input x is mapped by the ground truth g^* function to the motif map m^* of the positions of every digit/protein binding sites, which is itself mapped by the ground truth h^* function to the output y^*, the sequence `072634`/splice sites. Only x and y^* are available during training; the goal is to reconstruct g^* and h^*. Note that in splicing, unlike DIGITCIRCLE, the motifs can overlap. The M dots indicate the representation as described in Section 2, which is a one-hot encoding at each location (on the figure, each color indicates a different plane of the image, hence the last dimension being 10, for 10 digits).

the structure of the functions relating the motifs and the output (e.g., limited number of layers). Second, we describe the SPARLING algorithm, which allows for training models with an extreme sparsity constraint. We accomplish this via a layer that sets activations below some threshold equal to zero; this threshold is iteratively updated to achieve a target sparsity level (e.g., 99%). In order to address the unstable optimization landscape this produces at high sparsity values, our optimization algorithm anneals the target sparsity over time. Finally, we demonstrate several domains in which SPARLING can correctly identify the intermediate latent variable. These domains, while synthetic, demonstrate that the identifiability guarantee we proved is achievable in practice. In particular we present the highly synthetic DIGITCIRCLE domain as well as two more realistic domains: LATEX-OCR, in which we predict a LaTeX sequence from a noisy image of an algebraic expression, and AUDIOMNISTSEQUENCE, in which we predict a number from noisy audio of digits being spoken.

Related work. We summarize the related work here, and provide a more detailed discussion in Appendix A. Most existing work on learning interpretable latent representations assume some prior knowledge about the representations, including both concept bottleneck models and the Genomics work mentioned above. The recently proposed "Language in a Bottle" technique (Yang et al. (2023)) proposes to address this problem by using large language models (LLMs) to identify intermediate concepts; however, this is only applicable to certain domains. Finally, our theoretical work is connected to the statistical literature on identifiability, which asks whether the "true" parameters of a model can be recovered from data. Indeed, prior work has proposed algorithms that guarantee identification of latent variable models such as Hidden Markov Models (HMMs) (Yoon (2009)) and Probabilistic Context-Free Grammars (PCFGs) (Hsu et al. (2012)). While the problem is similar, we are interested in the deep learning setting where the latent concepts form the intermediate layer between two arbitrary models (presumably neural networks). Then, our theoretical results establish assumptions on the models and data distributions under which we can guarantee recovery of the "true function". This problem is similar to that of nonlinear Independent Component Analysis (ICA) (Hyvärinen et al. (2023); Khemakhem et al. (2020)), where the goal is identifying independent components mixed by some nonlinear function. However, we attempt to make much more limited assumptions of the "mixing function" and show that small end-to-end error is sufficient to imply recovery of the latent concepts. While our algorithm is not guaranteed to achieve small end-to-end error, this is a useful theorem as verifying low end-to-end error is trivial given a test set. Additionally, we find that in our experiments we do achieve low end-to-end error.

2 PROBLEM FORMULATION

We are interested in settings where intermediate activations represent latent variables corresponding to semantically meaningful concepts in the prediction problem. To this end, we consider the case

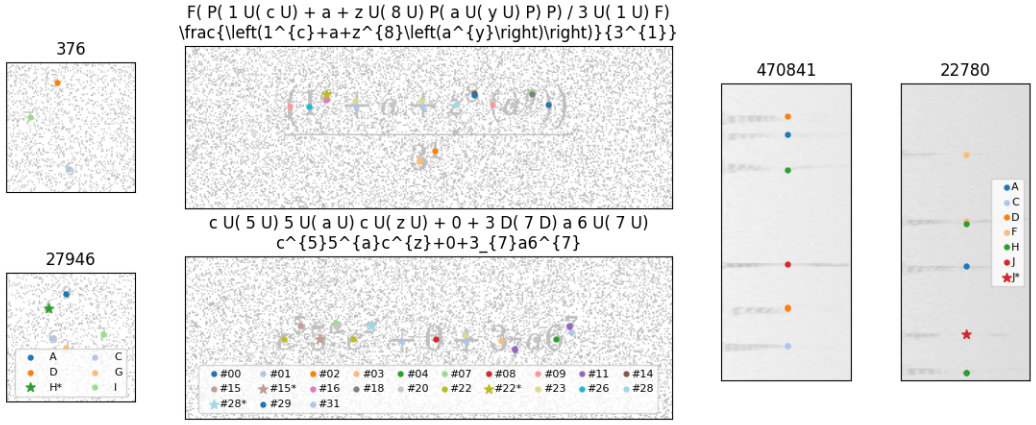

Figure 2: Two examples of inputs (images), outputs (sequences in titles), and our $\hat{g}$ predictions for seed=1 (colored dots) for DIGITCIRCLE, LATEX-OCR, and AUDIOMNISTSEQUENCE. For LATEX-OCR, we provide the output twice, first as the sequence of commands generated by the network and second as the translation of those commands into LaTeX. We place a dot for every maximal motif, colored/labeled by the channel that it appears in (e.g., the 0th channel is `A` or `#00`, 1st is `B` or `#01`, etc.). Stars indicate sites where non-maximal motifs are present as well.

where the *ground truth* is represented as a function $f^* : X \to Y$ composed $f^* = h^* \circ g^*$ of two functions $g^* : X \to M$ and $h^* : M \to Y$. We call the latent space M the *motif* space. We consider the task of training $\hat{g}$ and $\hat{h}$ to accurately model g^* and h^* using only end-to-end data $\mathcal{D} = \{(x, f^*(x)) : x \sim \mathcal{D}_X\}$ (i.e., enforcing only that their composition $\hat{f} = \hat{h} \circ \hat{g}$ accurately models f^*)[1]. Importantly, we assume no access to data on M (in particular, which components of M are active for any particular input). Our goal is to establish the conditions under which this task is possible and to present an algorithm to derive $\hat{g}$ and $\hat{h}$. Specifically, we focus on the case where g^* and $\hat{g}$ exhibit the properties of locality and sparsity as described below.

We assume that elements of $X \subseteq \mathbb{R}^{I \times [d]}$ and $M \subseteq \{0,1\}^{I \times [n]}$ are tensors, where $[d] = \{1, ..., d\}$, and where $I = [D_1] \times \ldots \times [D_l]$ is a set of spatial indices (e.g., for a 2D image, D_1 and D_2 would be the height and width of the image, respectively), d is the number of input channels (e.g., 3 for an RGB image or 4 for ACGT in one-hot encoded RNA), and n is the number of kinds of motif (e.g., for 10 for DIGITCIRCLE, with one for each digit, and 79 for splicing, with one for each protein type). In addition, Y is a discrete label space.

Locality We define the set $\mathcal{G}$ of "local" motif models as a generalization of convolutional models. Specifically, we want to have a definition that roughly corresponds to models whose output at each point is defined by a fixed number of inputs whose indices are determined independently of the actual values of the input. This property is most obviously present in convolutional layers, but also exists in, e.g., graph convolutions. SPARLING relies on locality to treat different parts of the input as independent, alongside the INPUT-FACTORIZATION assumption we introduce later.

Formally, we define the set $\mathcal{G}$ relative to some $t \in \mathbb{N}$ and "footprint function" $p : I \to I^t$ (in the case of a 2D convolution with kernel width w this would be a map from an index i in the output layer to the set of $t = w^2$ indices in a square around i). We then say $g \in \mathcal{G}$ if there exists some "local version" of g: $g_l \in \mathbb{R}^{t \times d} \to \{0,1\}^n$ such that $g(x)[i, c] = g_l(x[p(i)])[c]$; i.e., the output in position i can be computed by collecting the inputs in the region $p(i)$ and feeding them to a local function

[1]We do not consider noise for the purposes of this paper. The result could be modified to handle IID Bernoulli noise in the error function by replacing the end-to-end error with end-to-end error minus irreducible error in the theorem statement.

g_l. For example, in the case of convolution, g_l is the convolution kernel.[2] We also define the "motif cell" $p_2(i)$ as the set of indicies $i' \in I$ whose footprints overlap that of $i \in I$:[3]

$$p_2(i) = \{i' \in I : p(i) \cap p(i') \neq \emptyset\}$$

and define Δ such that for all $i \in I$, $p_2(i) \subseteq \{i + d : d \in \Delta\}$, e.g., if g is a 2D convolutional model with kernel size $(2r + 1, 2r + 1)$, then $\Delta = \{-2r, \ldots, 2r\}^2$. $|\Delta|$ appears in our error bound.

Sparsity Let the number of motifs for a channel c in a given activation pattern $m = g(x)$ be $\#_c(m) = \sum_{i \in I} \mathbf{1}(m[i, c] \neq 0)$. We can then define the mean value of this over the dataset for a given motif function as $\#_c(g) = \mathbb{E}_x[\#_c(g(x))]$. Let $\#(t) = \sum_c \#_c(t)$ for both elements of M and $\mathcal{G}$. Let the *density* of a model be $\delta(g) = \#(g)/|I \times [n]|$ and $\delta^* = \delta(g^*)$. We refer to $1 - \delta(g)$ as the *sparsity* of g.

3 Motif Identifiability Theorem

3.1 Theorem Statement

We define *Motif Identifiability* as a property of a data distribution $\mathcal{D}_X$ and mechanism g^*, h^*. Intuitively, it says that for any estimate $\hat{f} = \hat{h} \circ \hat{g}$ of f^*, if $\hat{f}$ has low end-to-end error, then $\hat{g}$ must have low motif error (i.e., $\hat{g}$ is a good estimate of g^*). In other words, if we are able to learn a model on (x, y^*) data that achieves good end-to-end error, then we can conclude that we have correctly estimated m^* even if we do not have any data on m^*. Formally, if Binarization, Non-Overlapping, Input-Factorization, and α-Motif-Importance (defined in Section 3.3) hold, then for some $k = O\left(\frac{\#_{\max}^2 |\Delta| n^2}{\#^* \alpha^2}\right)$, we have

$$\forall \hat{g} \in \mathcal{G} \,.\, \delta(\hat{g}) = \delta^* \implies \left(\forall \epsilon > 0, \mathcal{E}(\hat{h} \circ \hat{g}) < \epsilon \implies \mathcal{E}_m(\hat{g}) < k\epsilon\right)$$

where $\mathcal{E}$ is end-to-end error and $\mathcal{E}_m$ is motif error, as defined in Section 3.2.

For simplicity, we describe our error metrics and assumptions as if $n = 1$, that is, there is only one channel. We provide multi-channel versions of these formally in Appendix B.

3.2 Error Metrics

We define error metrics for both end-to-end error and motif error in two ways: a mathematically simple definition for our proof, and a more intuitive definition for our empirical findings (see Section 5.1). We demonstrate that these are equivalent modulo a constant factor in Appendix E. For our proofs, we define end-to-end error as exact match: $\mathcal{E}(\hat{f}) = \mathbb{E}_{x,y^* \sim \mathcal{D}}[\hat{f}(x) \neq y^*]$.

Defining the motif error metric, $\mathcal{E}_m$, is more complex. In particular, the definition of equivalence needs to account for $\hat{g}$ placing the motifs at slightly different locations, or permuting the motif channels. Thus, we only check that the predicted point be within the motif cell of a given true motif. In this section, we assume there is only one channel, so there is no channel permutation problem, but in Appendix B.1, we handle channel permutations by taking a minimum over all possibilities.

For our proofs, we define motif error using an intersection-over-union-inspired metric. For the "intersection" in this metric we use the number of true motif cells in $g^*(x)$ covered by a unique motif in $\hat{g}(x)$. To define this, we first define the function $v_{\hat{m}}(i)$ to be the number of motifs in the motif cell surrounding i in $\hat{m} = \hat{g}(x)$:

$$v_{\hat{m}}(i) = \sum_{i' \in p_2(i)} \mathbf{1}(\hat{m}[i'] \neq 0)$$

[2]Note: this notion of locality is more general than most, as $p(i)$ do not need to be "near" each other in the input space; however the Non-Overlapping assumption implies that $p(i)$ cannot simply be an arbitrary set. We do not restrict ourselves to convolutional locality; our definition could apply to e.g., neighbors on graphs.

[3]We use this notation because for convolutions with no max pooling, and ignoring the d axis, $p_2 = p \circ p$.

We then define $u(\hat{g}(x), g^*(x))$ to be the number of motif cells in the true motif pattern $g^*(x)$ that are covered by exactly one motif in the predicted motif pattern $\hat{g}(x)$.

$$u(\hat{m}, m^*) = \sum_{i \in I} \mathbf{1}\left(m^*[i] \neq 0 \wedge v_{\hat{m}}(i) = 1\right)$$

We then take the expectation of u over the dataset to get our "intersection" value. For our "union" value, we take the maximum of the expected number of motifs produced by g^* and $\hat{g}$: $\max(\#(\hat{g}), \#^*)$. The result is our metric

$$\mathcal{E}_m(\hat{g}) = 1 - \frac{\mathbb{E}_{x \sim \mathcal{D}}\left[\sum_{c'} u(\hat{g}(x), g^*(x))\right]}{\max(\#(\hat{g}), \#^*)}$$

This metric is directionally correct under all circumstances, rewarding $\hat{g}$ that produce motifs that overlap cells of g^* with a lower error[4].

3.3 Formal Assumptions

We assume our data generation process is represented by a graphical model $x \leftarrow m^* \rightarrow y^*$; intuitively, the motifs m^* are sampled first, and then x and y^* are sampled conditioned on m^*. This allows us to describe our assumptions as constraints on $P(x|m^*)$ and $P(y^*|m^*)$.

At a high level, we assume motifs cannot appear near each other (Non-Overlapping) and $P(x|m^*)$ must be easily decomposed into factors in order to constrain the relationship between x and m^*, ensuring that x is a product of distributions describing the footprints of motifs (Input-Factorization). This is our main assumption, analogous to a Markovian assumption in a Hidden Markov Model. Next, α-Motif-Importance describes the relationship between m^* and y^*, asserting that all motifs are important in some cases; in other words, h^* cannot systematically ignore any motif or treat any two motifs as interchangeable. This assumption ensures the definition of "motif" is restricted to concepts that are possible to learn from end-to-end data, analogous to a full-rank covariance assumption in Linear Regression.

While these constraints may appear strict, they fit problems where g^* identifies small local patterns in the input—e.g. motifs such as the individual digits in DigitCircle—that are all used at least sometimes by h^*. However, they do *not* fit the splicing domain (primarily Non-Overlapping and Input-Factorization), necessitating the additional data used by Gupta et al. (2024).

Binarization We assume that $\hat{g}$ is binary—i.e., $\hat{g}(x)[i, c] \in \{0, 1\}$ at all positions.

Non-Overlapping We assume that motif cells cannot overlap in samples drawn from $\mathcal{D}_X$

$$\forall x \in X, P_x(x) > 0 \implies \forall i, i' \in g^*(x), i \neq i' \implies p_2(i) \cap p_2(i') = \emptyset$$

Input-Factorization We assert that probability $P_x(x)$ decomposes to independent distributions for each patch $p(i)$ for which $g^*(x)[i] \neq 0$ and a background probability covering all non-patch inputs. Formally, we define the probability of x given that it produces the motif pattern m as

$$P[x|g^*(x) = m] = \left(\prod_{i \in m} P_f(x[p(i)])\right) P_b\left(x[r(m)]\right)$$

where $i \in m$ if $m[i] \neq 0$ and where P_f is a distribution over "foreground" parts of the input (those containing motifs) and $P_b(x)$ is a distribution over "background" x, and we are taking a marginal $x[r(m)]$, where $r(m) = I \times [d] \setminus \bigcup_{i \in m} p(i)$ is the set of all indices not in any motif footprint. We then represent $P[x] = \sum_{m \in M} P[x|g^*(x) = m] P_m(m)$ where $P_m(m)$ is our distribution over m.

We also require that P_b be translationally invariant. Specifically, for all sets $L \subseteq I$ and all offsets $o \in \mathbb{Z}^l$ such that $\{i + o : i \in L\} \subseteq I$, we have $P_b(x[L]) = P_b(x[\{i + o : i \in L\}])$. That is, the joint distribution should be the same at each location regardless of translation. This property holds for all datasets created by clipping random components of larger datasets, e.g., clipping sequences of RNA from the genome or snippets of text from a book. See Appendix F.1 for a motivating counterexample.

[4]If we assume Non-Overlapping we also have that $\mathcal{E}(g^*) = 0$

α-MOTIF-IMPORTANCE We wish to assert that no motif can be ignored for the purposes of computing the output in almost all inputs. Our assumption is parameterized by α; motif importance with a higher α implies that motif errors will lead to end-to-end errors on a higher fraction of the input. This assumption has a particularly complex formulation to ensure its weakness, and should not generally be the reason this theorem does not apply to a domain.

We begin by defining a *perturbation function* $R(m_1)$ that relates a motif m_1 to a set $m_2 \in R(m_1)$ which corresponds to m_1 with a motif deleted.

As a preliminary, we define STRONG-α-MOTIF-IMPORTANCE: here we require the existence of a pair m_1, m_2 where $m_2 \in R(m_1)$, $P_m(m_1), P_m(m_2) \geq \alpha$, and $h^*(m_1) \neq h^*(m_2)$. In this scenario, a model $\hat{g}$ that produces the same result on both m_1 and m_2 would produce the wrong answer on at least α of the dataset. Unfortunately, this assumption is far too strong to apply to any realistic domain since $P_m(m_1)$ and $P_m(m_2)$ will be very small as $|M|$ is exponentially large.

For α-MOTIF-IMPORTANCE we want to generalize the above notion such that the bound α applies to a set of m_1 values. Unfortunately, this is not as simple as computing the probability of a set

$$E \subseteq \{m_1 : \exists m_2 \in R(m_1), h^*(m_1) \neq h^*(m_2)\}$$

as we need to establish not only properties of $m_1 \in E$, and the corresponding m_2, but also allow for the fact that multiple m_1 might correspond to some m_2 (e.g., many m_1 that have a given motif at slightly different positions all correspond to the same m_2 once that motif is deleted). To resolve this, we replace E with a probability distribution $\psi_R(m_2|m_1)$ mapping m_1 to a distribution over $m_2 \in M \cup \{\perp\}$, where the $\perp$ represents no pairing. We assert that for this assumption to apply there must exist a ψ_R that is supported only on perturbations (m_1, m_2); formally:

$$\psi_R(m_2|m_1) > 0 \implies h^*(m_1) \neq h^*(m_2) \wedge m_2 \in R(m_1)$$

We can then define a process $q_R(m_2) = \sum_{m_1} P_m(m_1)\psi(m_2|m_1)$, that is, sample $m_1 \sim P_m$ then a perturbation $m_2 \sim \psi(m_2|m_1)$. We assert that $\forall m_2 \in M, q_R(m_2) \leq P_m(m_2)$, that is, this process can never lead to more probability mass on m_2 than the original distribution (see Appendix F.2 for a counterexample motivating this). Finally, we assert that $\sum_{m_2 \in M} q_R(m_2) = 1 - q_R(\perp) \geq \alpha$. [5]

Putting this all together, we have the following formal definition of α-MOTIF-IMPORTANCE: let $R : M \to 2^M$ such that $m_2 \in R(m_1)$ iff there exists $i \in I$ such that m_1 and m_2 agree except that $m_1[i] \neq 0$ and $m_2[p_2(i)] = 0$. Then, we assume the existence of some $\psi_R(m_2|m_1)$ such that

- $\forall m_1, m_2 \in M, \psi_R(m_2|m_1) > 0 \implies h^*(m_1) \neq h^*(m_2) \wedge R(m_1, m_2)$
- $\forall m_2 \in M, q_R(m_2) \leq P_m(m_2)$
- $\sum_{m_2 \in M} q_R(m_2) \geq \alpha$

3.4 PROOF SKETCH

We give a proof of this theorem in Appendix D. In short, we proceed by contrapositive, assuming high motif error. We then establish via a counting argument that since $\delta(\hat{g}) = \delta^*$, any motif error must either be due to false negatives or confusion (a channel of $\hat{g}$ being used for two different motifs). In both cases, we then establish that this error must apply to some fraction of all motif sites (via INPUT-FACTORIZATION), then establish that this should lead to a perturbation described in α-MOTIF-IMPORTANCE with some proportional probability, and thus to end-to-end error.

4 METHODS

SPARLING trains models with *Spatial Sparsity Layer*s using the *Adaptive Sparsity Algorithm*.

[5] In practice, for all of our synthetic domains, we can prove α-MOTIF-IMPORTANCE for high α (above 0.5). The perturbation process involves selecting a motif at random and deleting it, then rejecting with some probability (generally about 10%). This works because while several possible deletions on different m_1 can lead to the same m_2, the m_1s each have more degrees of freedom thus lower probability. This property will be shared by any dataset which contains a high-probability subset with a roughly uniform distribution over number of motifs, which is true for most datasets. The rejection outcome is necessary since spacing constraints mean that e.g., not all 3-digit DIGITCIRCLE tasks are representable as a deletion of a 4-digit task.

4.1 SPATIAL SPARSITY LAYER

This layer is the last step in the computation of $\hat{g}$ and enforces its sparsity. We define a spatial sparsity layer to be a layer with a parameter t whose forward pass is computed

$$\text{Sparse}_t(z) = \text{ReLU}(z - t)$$

Importantly, t is treated as a constant in backpropagation and is thus not updated by gradient descent. Instead, we update t using an exponential moving average of the quantiles of batches[6]:

$$t_n = \mu t_{n-1} + (1 - \mu) q(z_n, 1 - \delta),$$

where t_n is the value of t on the nth iteration, z_n is the nth batch of inputs to this layer, μ is the momentum (we use $\mu = 0.9$), δ is the target density, and $q : \mathbb{R}^{B \times d_1 \times \ldots \times d_k \times n} \times \mathbb{R} \to \mathbb{R}^n$ is the standard `torch.quantile` function. q is applied across all dimensions except the last: it produces a value for each channel that represents the threshold u for which the proportion of elements above u in the tensor at that channel is δ. We describe an alternative in Appendix J.3. Since t_n is fit to the data distribution, we can treat this as a layer that enforces that $\hat{g}$ has a sparsity of $1 - \delta$. Finally, we always include an affine batch normalization before this layer to increase training stability. We provide an analysis on the necessity of this addition in Appendix J.2.

4.2 ADAPTIVE SPARSITY ALGORITHM

Algorithm 1 Train Loop $(\hat{f}, \mathcal{D}, M, B, d_T, \delta_{\text{update}})$

```
T_0 ← 1
for t = 1 to ... do
  TRAINSTEP(f̂, D_{Bt:B(t+1)})
  T_t ← T_{t−1} − B d_T
  if bt mod M = 0 then
    A_t ← VALIDATE(f̂)
    if A_t > T_t then
      (f̂.δ, T_t) ← (f̂.δ × δ_update, A_t)
```

We found that applying an extreme sparsity requirement (very low δ) upon initial training of the network leads to the network getting stuck in a local minimum due to a lack of learning signal. To resolve this, we use a technique inspired by simulated annealing and reduce δ slowly over time. Annealing hyperparameters is a known technique (Sønderby et al. (2016), but we tie this annealing to validation accuracy (exact match between y^* and $\hat{f}(x)$) in order to be flexible to training schedule.

As shown in Algorithm 1, we add a step to our training loop that checks validation accuracy A_t and reduces the density whenever it exceeds a target T_t, reducing T_t over time. Our experiments use evaluation frequency $M = 2 \times 10^5$, batch size $B = 10$, $d_T = 10^{-7}$, and $\delta_{\text{update}} = 0.75$.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We describe our three new domains below. See Figure 2 for examples of each domain.

DIGITCIRCLE domain. The input x is a 100×100 monochrome image with 3-6 unique digits placed in a rough circular pattern, with some noise being applied to the image both before and after the numbers are placed. The output y^* is the sequence of digits in counterclockwise order, starting with the smallest number. The latent motifs layer m^* is the position of each digit: which can be represented as a $100 \times 100 \times 10$ tensor with 3-6 nonzero entries. Note that we have no access during training and validation to the concept of a digit as an image, nor to the concept of a digit's position.

LATEX-OCR domain. As a more realistic test, we take inspiration from Deng et al. (2016) and present the task of synthesizing LATEX code from images. This task is an OCR task like DIGITCIRCLE, but with variation in digit rendering (size, aliasing) and a more complex h^*.

[6]For numerical stability, we accumulate batches such that $|z_n|\delta \geq 10C$ before running this update

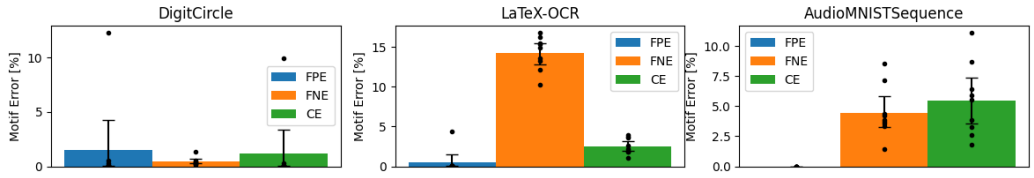

Figure 3: Motif Error, across three different metrics. Bar height depicts the mean across 9 seeds, individual dots represent seed, the error bar represents a 95% bootstrap CI. AUDIOMNISTSEQUENCE has an FPE of exactly 0. High FNE on LATEX-OCR is due to fraction bars, parentheses, and plus signs not being recognized in all cases since it is possible to infer the output without access to these. For a comparison of our technique to less-sparse models, see Figure 4.

AUDIOMNISTSEQUENCE domain. In this domain, we synthesize short clips of audio representing sequences of 5-10 digits over a bed of noise. The task is to predict the sequence of characters spoken. Here, we test if motif models can generalize: we train and validate with AUDIOMNIST (Becker et al. (2018)) samples from Speakers 1-51 and test with samples from Speakers 52-60.

Splicing domain. We also considered the splicing domain discussed in Gupta et al. (2024). Since it does not satisfy our assumptions from Section 3.3, SPARLING is not able to precisely identify the motifs, but does perform substantially better than random chance. See Appendix L for our results.

Architecture and training. Our neural architecture is adapted from that of Deng et al. (2016). For DIGITCIRCLE, we make $\hat{g}$ have a 17×17 overall window, by layering four residual units (He et al. (2016)), each containing two 3×3 convolutional layers. We then map to a 10-channel bottleneck where our Spatial Sparsity layer is placed. Our $\hat{h}$ architecture is a max pooling, followed by a similar architecture to Deng et al. (2016). We keep the LSTM row-encoder, but replace the attention decoder with a column-based positional encoding followed by a Transformer (Vaswani et al. (2017)) whose encoder and decoder have 8 heads and 6 layers. Throughout, except in the bottleneck layer, we use a width of 512 for all units. For LATEX-OCR we use the same architecture but with 32 motifs (to account for the additional characters) and a 65×65 overall window (to account for the larger characters, though we find 33×33 does not change the results substantially). For AUDIOMNISTSEQUENCE we process the audio via a spectrogram with a sample rate of 8000 and 64 channels, use a 33-wide 1D resnet stack for $\hat{g}$ and a transformer for $\hat{h}$. We generate training, validation, and test sets randomly. For efficiency, LATEX-OCR is looped on 10^7 training samples, the rest are infinite. We use a batch size of 10 and a learning rate of 10^{-5}. Our validation and test sets both contain 10^4 examples. Details on computational usage are in Appendix M.

Error Metrics For our empirical analysis, we use more granular error metrics, defining define end-to-end error as normalized edit distance:

$$\text{E2EE}(\hat{f}) = \mathbb{E}_{x,y^* \sim \mathcal{D}} \left[\frac{\text{EDITDISTANCE}(y^*, \hat{f}(x))}{\max(|y^*|, |\hat{f}(x)|)} \right].$$

and disaggregating motif error's false positives, false negatives, and mis-identified motifs into three separate metrics into False Positive (FPE), False Negative (FNE), and Confusion Error (CE) (confusion error occurs when multiple motif channels are confused, this is always zero if $n = 1$). Appendix C.2 contains formal definitions of these metrics and Appendix E contains a proof that these metrics are bounded within a constant multiplicative factor of $\mathcal{E}_m$.

5.2 RESULTS

Motif error. We show our three metrics of motif error in Figure 3 for each of our models on each domain. Motif errors for our model average below 10% for all our domains, except in the case of FNE on LATEX-OCR. The generally low motif errors, despite only training and validating end-to-end, demonstrate that our algorithm achieves Motif Identifiability on all three domains. This property even holds when generalizing to unseen samples in the AUDIOMNISTSEQUENCE experiment, providing evidence that SPARLING is genuinely learning the motif features rather than memorizing. The

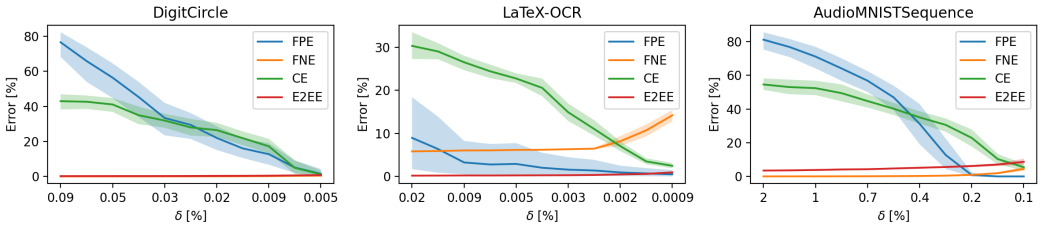

Figure 4: Motif and end-to-end error metrics versus δ. Note that the x axis is a reversed log-scale, since the adaptive sparsity algorithm starts with high density and narrows it exponentially.

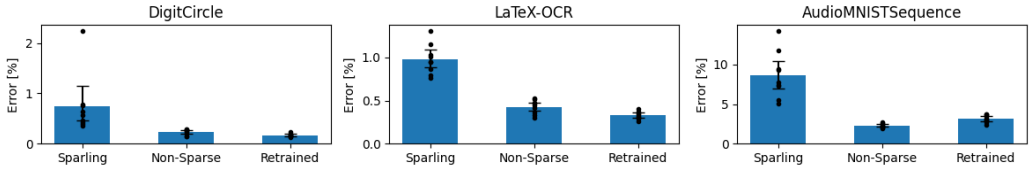

Figure 5: *Retrained* tends to perform as well as or slightly worse than *Non-Sparse*, making up most of the gap from SPARLING. The apparent improvement from *Non-Sparse* to *Retrained* should not be interpreted as real, the numerical difference is tiny and the sample accuracies overlap.

one case where our model has high error, FNE on LATEX-OCR, demonstrates the importance of the α-MOTIF-IMPORTANCE assumption: recognizing LATEX text in the space we generated does not require identification of fraction bars or all of `()+`. For more details, see Figure 2 and Appendix G. Interestingly, this only affects the unimportant digits; this is because our proof is still (mostly[7]) valid if some motifs are never used: they can simply be treated as part of the background instead.

Examples. Figure 2 shows a few examples for one of our models' intermediate layers. As can be seen, all digits are appropriately identified by our intermediate layer, with very few dots (in these examples, none) falling away from a digit. Note that the activations are consistent from sample to sample—for example, in DIGITCIRCLE, motif C is used for digit 6 in both images.

Necessity of Extreme Sparsity Figure 4 shows our error metrics plotted against the sparsity, with the x-axis reversed to show progression in training time as we anneal δ. As expected, as δ decreases, FPE decreases and FNE increases. More interestingly, we note a trade-off between E2EE and CE: as δ decreases, E2EE increases and CE decreases substantially. This demonstrates a trade-off between a more accurate overall model, which benefits from greater information present and a more accurate motif model, which benefits from a tighter entropy bound. Furthermore, CE is often substantially higher for even a 2-3$\times$ increase in δ, demonstrating the need for extreme sparsity. This validates the Motif Identification Theorem, which relies on $\delta(\hat{g}) = \delta^*$ to make its guarantees.

End-to-End error As seen in Figure 5, SPARLING tends to produce higher end-to-end errors than a baseline Non-Sparse model. We theorize that this is because our constraint on the information flow requires the model to "commit" to a choice on whether or not a given site is a true motif. To verify this effect, we present the *Retrained* setting, in which we remove the bottleneck, freeze the motif model $\hat{g}$, and finetune $\hat{h}$ on the training set until convergence. The Retrained setting tends to perform similarly to the Non-Sparse setting. We thus demonstrate that we are not degrading end-to-end performance unacceptably, even while substantially improving interpretability.

[7]The INPUT-FACTORIZATION assumptions regarding P_b are broken instead, but these are less crucial.

6 LIMITATIONS

This work applies only to data distributions with the properties we describe in Section 3.3. In practice, the main limiting assumption is INPUT-FACTORIZATION, which requires that the dataset is composed of several small independent patches. While this applies to many problems, it does not apply to problems that identify properties of a single coherent subject, e.g., the classic image classification tasks MNIST (Deng (2012)) and ImageNet (Deng et al. (2009)).

7 CONCLUSION

We prove that Motif Identification is solvable under certain assumptions. Additionally, we demonstrate SPARLING, a practical algorithm to learn end-to-end models that have a sparse intermediate layer. Finally, we demonstrate that Motif Identifiability is not solely theoretical: SPARLING achieves interpretable and accurate motifs with zero direct supervision on the motifs across three domains.

REFERENCES

Kartik Ahuja, Jason S Hartford, and Yoshua Bengio. Weakly supervised representation learning with sparse perturbations. *Advances in Neural Information Processing Systems*, 35:15516–15528, 2022.

Yusuf Aytar, Lluis Castrejon, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Cross-modal scene networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2303–2314, 2017.

Sören Becker, Marcel Ackermann, Sebastian Lapuschkin, Klaus-Robert Müller, and Wojciech Samek. Interpreting and explaining deep neural networks for classification of audio signals. *CoRR*, abs/1807.03418, 2018.

Paschalis Bizopoulos and Dimitrios Koutsouris. Sparsely activated networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(3):1304–1313, 2020.

Joachim Bona-Pellissier, François Bachoc, and François Malgouyres. Parameter identifiability of a deep feedforward relu neural network. *Machine Learning*, 112(11):4431–4493, 2023.

Hervé Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4):291–294, 1988.

Jack Brady, Roland S Zimmermann, Yash Sharma, Bernhard Schölkopf, Julius Von Kügelgen, and Wieland Brendel. Provably learning object-centric representations. In *International Conference on Machine Learning*, pp. 3038–3062. PMLR, 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477.

Yuntian Deng, Anssi Kanervisto, and Alexander M Rush. What you get is what you see: A visual markup decompiler. *arXiv preprint arXiv:1609.04938*, 10:32–37, 2016.

Guillaume Desjardins, Aaron Courville, and Yoshua Bengio. Disentangling factors of variation via generative entangling. *arXiv preprint arXiv:1210.5474*, 2012.

James Enouen and Yan Liu. Sparse interaction additive networks via feature interaction detection and sparse selection. *Advances in Neural Information Processing Systems*, 35:13908–13920, 2022.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Kavi Gupta, Chenxi Yang, Kayla McCue, Osbert Bastani, Phillip A Sharp, Christopher B Burge, and Armando Solar-Lezama. Improved modeling of rna-binding protein motifs in an interpretable neural model of rna splicing. *Genome Biology*, 25(1):23, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2016.

Daniel J Hsu, Sham M Kakade, and Percy S Liang. Identifiability and unmixing of latent parse trees. *Advances in neural information processing systems*, 25, 2012.

Aapo Hyvärinen, Ilyes Khemakhem, and Hiroshi Morioka. Nonlinear independent component analysis for principled disentanglement in unsupervised deep learning. *Patterns*, 4(10), 2023.

Kishore Jaganathan, Sofia Kyriazopoulou Panagiotopoulou, Jeremy F McRae, Siavash Fazel Darbandi, David Knowles, Yang I Li, Jack A Kosmicki, Juan Arbelaez, Wenwu Cui, Grace B Schwartz, et al. Predicting splicing from primary sequence with deep learning. *Cell*, 176(3): 535–548, 2019.

Neil Jethani, Mukund Sudarshan, Yindalon Aphinyanaphongs, and Rajesh Ranganath. Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In *International Conference on Artificial Intelligence and Statistics*, pp. 1459–1467. PMLR, 2021.

Nan Jiang, Wenge Rong, Baolin Peng, Yifan Nie, and Zhang Xiong. An empirical analysis of different sparse penalties for autoencoder in unsupervised feature learning. In *2015 international joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2015.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International conference on artificial intelligence and statistics*, pp. 2207–2217. PMLR, 2020.

Diederik P Kingma and Max Welling. Stochastic gradient vb and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, volume 19, pp. 121, 2014.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pp. 5338–5348. PMLR, 2020.

Sébastien Lachapelle, Pau Rodriguez, Yash Sharma, Katie E Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ica. In *Conference on Causal Learning and Reasoning*, pp. 428–484. PMLR, 2022.

Sébastien Lachapelle, Tristan Deleu, Divyat Mahajan, Ioannis Mitliagkas, Yoshua Bengio, Simon Lacoste-Julien, and Quentin Bertrand. Synergies between disentanglement and sparsity: Generalization and identifiability in multi-task learning. In *International Conference on Machine Learning*, pp. 18171–18206. PMLR, 2023.

Ismael Lemhadri, Feng Ruan, Louis Abraham, and Robert Tibshirani. Lassonet: A neural network with feature sparsity. *The Journal of Machine Learning Research*, 22(1):5633–5661, 2021.

Susan E Liao, Mukund Sudarshan, and Oded Regev. Machine learning for discovery: deciphering rna splicing logic. *bioRxiv*, pp. 2022–10, 2022.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

Rongrong Ma, Jianyu Miao, Lingfeng Niu, and Peng Zhang. Transformed l1 regularization for learning sparse deep neural networks. *Neural Networks*, 119:286–298, 2019. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2019.08.015. URL https://www.sciencedirect.com/science/article/pii/S0893608019302321.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. *Advances in neural information processing systems*, 27, 2014.

Gemma E Moran, Dhanya Sridhar, Yixin Wang, and David M Blei. Identifiable deep generative models via sparse decoding. *arXiv preprint arXiv:2110.10804*, 2021.

Andrew Ng. Cs294a lecture notes: Sparse autoencoder, Winter 2011. URL https://web.stanford.edu/class/cs294a/sparseAutoencoder.pdf.

Qihan Ren, Jiayang Gao, Wen Shen, and Quanshi Zhang. Where we have arrived in proving the emergence of sparse interaction primitives in dnns. In *The Twelfth International Conference on Learning Representations*, 2024.

Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2017.02.029. URL https://www.sciencedirect.com/science/article/pii/S0925231217302990.

Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

Sofia Serrano and Noah A Smith. Is attention interpretable? *arXiv preprint arXiv:1906.03731*, 2019.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. *Advances in neural information processing systems*, 29, 2016.

Yiyang Sun, Zhi Chen, Vittorio Orlandi, Tong Wang, and Cynthia Rudin. Sparse and faithful explanations without sparse models. *arXiv preprint arXiv:2402.09702*, 2024.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Alex M Tseng, Gökcen Eraslan, Nathaniel Lee Diamant, Tommaso Biancalani, and Gabriele Scalia. A mechanistically interpretable neural-network architecture for discovery of regulatory genomics. In *ICLR 2024 Workshop on Machine Learning for Genomics Explorations*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Danru Xu, Dingling Yao, Sébastien Lachapelle, Perouz Taslakian, Julius von Kügelgen, Francesco Locatello, and Sara Magliacane. A sparsity principle for partially observable causal representation learning. *arXiv preprint arXiv:2403.08335*, 2024.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19187–19197, 2023.

Byung-Jun Yoon. Hidden markov models and their applications in biological sequence analysis. *Current genomics*, 10(6):402–415, 2009.

Mateo Espinosa Zarlenga, Pietro Barbiero, Gabriele Ciravegna, Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, Zohreh Shams, Frederic Precioso, Stefano Melacci, Adrian Weller, et al. Concept embedding models. *arXiv preprint arXiv:2209.09056*, 2022.

Yujia Zheng, Ignavier Ng, and Kun Zhang. On the identifiability of nonlinear ica: Sparsity and beyond. *Advances in neural information processing systems*, 35:16411–16422, 2022.

Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. In *International conference on machine learning*, pp. 4140–4149. PMLR, 2017.

A Additional Related Work

Learning RNA/DNA motifs In Gupta et al. (2024) the authors introduce the concept of Sparse Adjusted Motifs. Specifically, they model the problem of splicing as a two stage process, in which first proteins bind the RNA sequence, and then cause the sequence to be spliced at certain points. Using end-to-end data of a sequence annotated with splicepoints, as well as baseline models of protein binding patterns in RNA, they are able to improve these models of protein binding. To accomplish this they use the baseline model to predict protein binding affinity, then apply SPARLING, a sparse layer, with a sparsity of $1 - 2\delta$. They then modify this with a neural network trained residually, allowing it to only influence nonzero sites, then apply another SPARLING layer with sparsity $1 - \delta$. In this work, we eschew the complexity off the Adjusted Motif model and instead consider the sparse layer by itself. In Tseng et al. (2024) and Liao et al. (2022), the authors learn motifs without intermediate supervision, but in these cases they heavily restrict the model class of the motif models, requiring them to be 1-layer convolutions.

Concept bottleneck models. Previous work also learns models with intermediate features that correspond to known variables. Some techniques, such as Concept Bottleneck Models (Koh et al. (2020)) and Concept Embedding Models (Zarlenga et al. (2022)), involve additional supervision with existing feature labels. Other techniques, such as Cross-Model Scene Networks (Aytar et al. (2017)), use multiple datasets with the same intermediate representation. The Language in a Bottle technique (Yang et al. (2023)) uses LLMs to identify intermediate concepts; however this is only applicable to certain domains (e.g., asking an LLM to produce the protein binding motifs in an RNA sequence will result in it providing a list of motif finding tools, not motifs). In this work, we do not require the presence of additional datasets or annotations.

Identifiability The problem of identifiability, in which the behavior of some component of a function is inferred via the behavior of the overall function, under some assumptions, is typically set up as an attempt to infer the values of specific parameters up to some isomorphism. In Hsu et al. (2012) the parameters are those of a PCFG expressing a distribution over sequences and the behavior of the function is the computation of a moment of this distribution (with infinite data). In Bona-Pellissier et al. (2023) the parameters are those of a multi layer ReLU network, identifiability is established with infinite data under several assumptions relating to the network as a piecewise linear function. Other work such as Zhong et al. (2017) focuses on strong convexity guarantees on the neighborhood of the true parameters, which is a far stronger claim as it leads to plausible inference algorithms; though the model class is restricted to 1 layer neural networks. In Ahuja et al. (2022), the result of sparse perturbations (perturbations of only some variables) to the latent variables is given, which enables identifiability; this differs from our α-MOTIF-IMPORTANCE in that we only assume the existence of perturbations that affect the observable output, rather than requiring access to these modified outputs as part of the dataset. In our case, we are attempting to infer the motif function $\hat{g}$ rather than any particular parameter, also up to isomorphism. As a result, we make weaker architectural assumptions about $\hat{g}$ and $\hat{h}$. However, the property we attempt to establish is stronger than identifiability with infinite data: we wish to show that the error in identification of $\hat{g}$ is bounded by a multiple of the end-to-end error. While this does not immediately lead to an inference algorithm, it implies that any inference algorithm that preserves our sparsity constraint while achieving low error will be a valid algorithm for identifying the true g^*.

Neural Input Attribution. SPARLING is useful for identifying the relevant parts of an input. One existing technique that accomplishes this goal is saliency mapping (Simonyan et al. (2013); Selvaraju et al. (2016)), which uses gradient techniques to find which parts of the input affect the output

most. Another technique, analyzing the attention weights of an attention layer (Mnih et al. (2014)), only works with a single layer of attention and does not necessarily produce valid or complete explanations (Serrano & Smith (2019)). Additionally, Amortized Explanation Techniques produce a subset of features that form a "local explanation," i.e., features sufficient to produce a prediction (Jethani et al. (2021)). The main benefit a sparse annotation provides over these techniques is unconditional independence: when using sparsity, you have the ability to make the claim "region $x[r]$ of the input is not relevant to the output prediction, regardless of the rest of the input $x[\bar{r}]$". This is a direct result of sparsity and locality and is unavailable when using saliency or attention techniques which inherently condition on the values you provide for $x[\bar{r}]$. Techniques such as Sparse Explanation Values (Sun et al. (2024)) do not have this guarantee, and so while they apply to a wider variety of model structures, they can thus only reason about local perturbations, providing local explanations of changes in behavior.

Disentangled representations. *Disentangled representations* are ones where different components of the representation encode independent attributes of the underlying data (Desjardins et al. (2012); Higgins et al. (2016)). Locatello et al. (2019) suggests there are no universal solutions to this problem, and all attempts require some prior about the kinds of representations being disentangled. We focus here on a prior regarding sparsity and locality.

Informational bottleneck. Other work also constrains the information content of the intermediate representation in a neural network. Strategies include constraining the dimension of the representation—e.g., PCA and autoencoders with low-dimensional representations Bourlard & Kamp (1988), or adding noise—e.g., variational autoencoders Kingma & Welling (2014). However, these approaches often encourage entangling features to communicate them through a smaller number of channels, and as such do not always learn interpretable representations of an intermediate state.

Sparse activations. Note that this notion of sparsity differs from *sparse parameters* Tibshirani (1996); Scardapane et al. (2017); Frankle & Carbin (2018); Ma et al. (2019); Lemhadri et al. (2021); Lachapelle et al. (2023), *sparse causal graphs* Moran et al. (2021); Lachapelle et al. (2022); Enouen & Liu (2022); Ren et al. (2024), and *sparse jacobians* Zheng et al. (2022); Brady et al. (2023); instead this line of work attempts to constrain the information content of an intermediate representation by encouraging *sparse activations*—i.e., each component of the representation is zero for most inputs. Sparse parameters serve different objectives and require different strategies to be used effectively. As sparse parameters only provide interpretability for single or two-layer models, they are generally used for efficiency in larger models. In terms of imposing sparsity, different techniques must again be used as sparse activation patterns depend on the input, so occasional pruning—e.g., Frankle & Carbin (2018)—is insufficient. Strategies for achieving sparse activations include imposing an L_1 penalty on the representation or a penalty on the KL divergence between the representation's distribution and a low-probability Bernoulli distribution Jiang et al. (2015). However, these techniques typically only achieve 50%-90% sparsity, whereas SPARLING can achieve $> 99.9\%$. We directly compare with these in Appendix J.1. Additionally, Bizopoulos & Koutsouris (2020) uses a quantile-based activation limit equivalent to both of our ablations (see Appendix J.2) combined, but in the simpler context of linear $\hat{h}$ and $\hat{g}$ models. Similarly, Xu et al. (2024) provides an identifiability result given sparse activations, but in the context of an affine model, whereas we allow arbitrary nonlinearity.

B MULTIPLE CHANNELS

We have to modify several definitions to handle the case of multiple channels. However, none of these changes modify the fundamental character of the theorem. Our provided proof (Appendix D) is for the more general case.

One useful definition is that of the set of true motifs: we define $\omega_c(m)$ to be a set of indices corresponding to motif of channel c: $\omega_c(m) = \{i \in I : \exists c, m[i, c] \neq 0\}$, we have that $i \in m \iff \exists c, i \in \omega_c(m)$.

B.1 MOTIF ERROR

Since there are multiple channels, and there is no way for $\hat{g}$ to know *a priori* what the appropriate assignment of motif to channel is, the predicted motifs models should be deemed equivalent to the ground truth model —which is known when we test—if there exists a channel assignment for which they are equivalent.

We follow a similar metric to the one described in Section 3.2 except channel-specific and then minimized over all assignments $\tau : [n] \to [n]$ of channels of $\hat{g}$ to channels in g^*:[8].

Our definition of v is modified by identifying a channel

$$v_{\hat{m}}(i, c') = \sum_{i' \in p_2(i)} \mathbf{1}(\hat{m}[i, c'] \neq 0)$$

We then modify u to be the number of motif cells of channel c in the true motif pattern $g^*(x)$ that are uniquely covered by a motif of channel c'.

$$u(\hat{m}, m^*, c', c) = \sum_{i \in I} \mathbf{1}\left(m^*[i, c] \neq 0 \wedge v_{\hat{m}}(i, c') = 1 \wedge \forall c'' \neq c', v_{\hat{m}}(i, c'') = 0\right)$$

We then take the sum of this over all channels c' of $\hat{g}$ and corresponding channels $\tau(c')$ of g^*, and then take an expectation over the dataset to get our "intersection" value, the expected number of true motif cells covered by a unique predicted motif of the corresponding channel.

$$\mathbb{E}_{x \sim \mathcal{D}}\left[\sum_{c'} u(\hat{g}(x), g^*(x), c', \tau(c'))\right]$$

Our union value is unchanged, leading to the metric:

$$\mathcal{E}_m(\hat{g}) = \min_{\tau}\left(1 - \frac{\mathbb{E}_{x \sim \mathcal{D}}\left[\sum_{c'} u(\hat{g}(x), g^*(x), c', \tau(c'))\right]}{\max(\#(\hat{g}), \#^*)}\right)$$

B.2 INPUT-FACTORIZATION ASSUMPTION

We slightly modify our assumption to

$$P[x | g^*(x) = m] = \left(\prod_{c \in [n], i \in \omega_c(m)} P_c(x[p(i)])\right) P_b\left(x[r(m)]\right)$$

which is identical except that P_f is replaced by a P_c, which is distinct for each channel.

B.3 α-MOTIF-IMPORTANCE ASSUMPTION

Adding more channels means we need to consider multiple perturbation functions. Specifically, we consider two classes of *perturbation functions* $R(m_1)$ that relate pairs of motif patterns, those where $m_2 \in R(m_1)$ correspond to m_1 with motif of a particular channel c_1 deleted, and those where $m_2 \in R(m_1)$ corresponds to m_1 with a particular motif of channel c_1 mutated into a motif of channel c_2.

One additional subtlety is that we also require flexibility to shifts in the perturbed motif's position, ensuring that the precise positions of motifs are not determined by the rest of the motifs, precluding a situation where, e.g., motifs are aligned to a grid, and the learned $\hat{g}$ "sneaks through" information about a motif's channel via off-grid positioning. Note that this means α-MOTIF-IMPORTANCE is tied to locality and in particular only makes sense when g^* is local within Δ as defined in Section 2.

Our formal definition then is over $\mathcal{R}$, a set of perturbation relations defined as follows:

[8]We do not require τ to be a permutation, as in practice we might want to allow extra channels in case we do not know the exFor our proofs, we define motif error using an intersection-over-union-inspired metricact number. In this case, the metric will only provide a good score if a real g^* motif is split up among channels of $\hat{g}$, but not if a single channel c' of $\hat{g}$ corresponds to multiple channels of g^*, which would be a loss of information

- For all $d_1 \in \Delta$ and c_1 let there be some $R \in \mathcal{R}$ such that $m_2 \in R(m_1)$ if and only if there exists some $i \in I$ such that m_1 and m_2 agree everywhere except that $m_1[i + d_1][c_1] \neq 0$ and $m_2[p_2(i)] = 0$
- For all $d_1, d_2 \in \Delta$ and $c_1 \neq c_2$ let there be some $R \in \mathcal{R}$ such that $m_2 \in R(m_1)$ if and only if there exists some $i \in I$ such that m_1 and m_2 agree everywhere except that $m_1[i + d_1][c_1] \neq 0$ and $m_2[i + d_2][c_2] \neq 0$

Then for each $R \in \mathcal{R}$ we assert the existence of some ψ_R such that our properties hold.

C Evaluation Metric Details

C.1 Preliminaries

We now define our FPM and MM motif sets, along with the C function.

Predicted motifs. For a given predicted motif tensor $\hat{m}$, we define $P(\hat{m}) = \{(i, c') : \hat{m}[i, c'] > 0\}$ to be the set of motifs predicted in $\hat{m}$, where $i \in I, c \in [n]$. Typically, we are interested in the set of motifs $P(\hat{g}(x))$ for our estimated motif model $\hat{g}$.

Footprint identification. Let $C : I \times M \to I \cup \{\perp\}$ be a function that identifies the motif cell that a given index is within, or $\perp$ otherwise:

$$C(i', m^*) = i \iff i' \in p_2(i)$$

By Non-Overlapping, this is always unique, but we can extend the definition to be coherent otherwise by giving it flexibility to choose an arbitrary such i:

$$(C(i', m^*) = i \impliedby i' \in p_2(i)) \wedge (C(i', m^*) = \perp \iff \forall i, i' \notin p_2(i))$$

False Positive Motifs. We now have the ability to define our first class of motifs: *false positive motifs*. These are predicted motifs that do not correspond to any real motifs:

$$\mathrm{FPM}(\hat{m}, m^*) = \{(i', c') \in P(\hat{m}) : C(i', m^*) = \perp\}.$$

We denote the remaining motifs by

$$P_1(\hat{m}, m^*) = P(\hat{m}) \setminus \mathrm{FPM}(\hat{m}, m^*).$$

Maximal Motifs First, we need to define the set of all predicted motifs that cover the same footprint as a given predicted motif. We do so via the $A_{\hat{m}, m^*}$ function, which takes a given predicted motif (assumed to overlap some footprint) and returns all others covering the same footprint:

$$A_{\hat{m}, m^*}(i', c) = \{(i'', c') \in P(\hat{m}) : C(i'', m^*) = C(i', m^*)\}$$

Now we can define *maximal motifs* are predicted motifs that are maximal in the footprint they cover:

$$\mathrm{MM}(\hat{m}, m^*) = \{t \in P_1(\hat{m}, m^*) : \hat{m}[t] = \max_{t' \in A_{\hat{m}, m^*}(t)} \hat{m}[t']\}$$

We can also define *non-maximal motifs* are predicted motifs that are non-maximal in the footprint they cover:

$$\mathrm{NMM}(\hat{m}, m^*) = \{t \in P_1(\hat{m}, m^*) : \hat{m}[t] \neq \max_{t' \in A_{\hat{m}, m^*}(t)} \hat{m}[t']\}$$

However, we ignore non-maximal motifs entirely for the purposes of our analysis, under the reasoning that these are trivially removable in practice.

C.2 Motif Error Metric

We then define three motif error metrics that we use empirically in evaluating our learned $\hat{g}$ models.

First, the *false positive error (FPE)* is the percentage of motifs that are false positive motifs.

$$\mathrm{FPE}_{\mathcal{D}}(\hat{g}) = \frac{\mathbb{E}_{x \sim \mathcal{D}}[|\mathrm{FPM}(\hat{g}(x), g^*(x))|]}{\mathbb{E}_{x \sim \mathcal{D}}[|P(\hat{g}(x))|]}.$$

Second, the *false negative error (FNE)* is the percentage of true sites that are not covered by any motif.

$$\mathrm{FNE}_{\mathcal{D}}(\hat{g}) = \frac{\mathbb{E}_{x\sim\mathcal{D}}[|\{(i,c) \in P(g^*(x)) : \nexists (i',c') \in P(\hat{g}(x)) : i' \in p_2(i)\}|]}{\mathbb{E}_{x\sim\mathcal{D}}[|P(g^*(x))|]}.$$

Finally, the *confusion error (CE)* is defined as follows: (i) rearrange $\hat{g}$'s channels to best align them with g^*, (ii) compute the percentage of maximal motifs in footprint of a true motif that do not correspond to the true motif's channel:

$$\mathrm{CE}_{\mathcal{D}}(\hat{g}) = \min_{\tau:[n]\to[n]} \frac{\mathbb{E}_{x\sim\mathcal{D}}[|\mathrm{conf}_\tau(\hat{g}(x), g^*(x))|]}{\mathbb{E}_{x\sim\mathcal{D}}[|\mathrm{MM}(\hat{g}(x), g^*(x))|]},$$

$\mathrm{conf}_\tau(\hat{m}, m^*)$ represents the motifs that do not match ground truth under rearrangement τ

$$\mathrm{conf}_\tau(\hat{m}, m^*) = \{t \in \mathrm{MM}(\hat{m}, m^*) : \neg \mathrm{mat}_\tau(t, C(t, m^*))\}|$$

and $\mathrm{mat}_\tau(t, t^*)$ is a function that checks whether the two motif index tuples match under channel rearrangement τ.

A low FPE/FNE implies that the model is identifying relevant portions of the input, while a low CE implies that the model classifies these components as motifs correctly.

D Proof of Motif Identifiability Theorem

The following is a formal proof of the Motif Identifiability Theorem. The term $\#^*_{\max}$ is used in this proof to denote $\max_c \#^*_c$

D.1 Proof Sketch

We proceed by contrapositive, starting with the assumption that $\mathcal{E}_m(\hat{g}) \geq k\epsilon$ and then proving that $\mathcal{E}(\hat{h} \circ \hat{g}) \geq \epsilon$. We first demonstrate (Lemma D.2.1) that high motif error implies either a high number of false negatives for some channel c (true motif cells that have no coverage by $\hat{g}$) or simultaneously a low number of false positives and some channels c_1, c_2 such that there is some high number of cells of both that are covered by the same channel c' of $\hat{g}$. This theorem is proven by a simple counting argument, relying only on the fact that $\delta(\hat{g}) = \delta^*$. We then prove in each of the two resulting cases that the property holds for some fraction of motif cells in general, using Input-Factorization and Non-Overlapping. We then apply α-Motif-Importance to each case, demonstrating that $\hat{g}$ does not distinguish different inputs (this argument uses Binarization) that must lead to different values of $y^* = h^*(g^*(x))$. Since $\hat{g}$ cannot distinguish these inputs, neither can $\hat{h} \circ \hat{g}$, and thus in one of the two cases error must arise. Thus, in both cases, we conclude that $\mathcal{E}(\hat{h} \circ \hat{g}) \geq \epsilon$.

D.2 Lemmas

D.2.1 Sources of Motif Error Dichotomy

First, we define a few quantities, representing the number of times a true cell is covered negatively, covered once, or covered multiple times.

$$\mathrm{FN}_g(c) = \mathbb{E}_{x,m^*}\left[\sum_{i\in\omega_c(m^*)} \mathbf{1}(v_{\hat{g}(x)}(i) = 0)\right]$$

$$\mathrm{CO}_g(c,c') = \mathbb{E}_{x,m^*}\left[\sum_{i\in\omega_c(m^*)} \mathbf{1}\left(v_{\hat{g}(x)}(i) = 1 \wedge v_{\hat{g}(x)}(i,c') = 1\right)\right]$$

$$\mathrm{CM}_g(c) = \mathbb{E}_{x,m^*}\left[\sum_{i\in\omega_c(m^*)} \mathbf{1}\left(v_{\hat{g}(x)}(i) > 1\right)\right]$$

We also define the quantity

$$\text{FP}_g = \mathbb{E}_{x,m^*}\left[\text{FP}_g(x)\right]$$

where

$$\text{FP}_g(x) = \sum_{c'} \sum_{i \in I \setminus \bigcap_c \omega_c(m^*)} \mathbf{1}(\hat{g}(x)[i,c'] = 1)$$

The claim we wish to establish is

$$\begin{aligned}
&\forall \hat{h} \in M \to Y, \hat{g} \in \mathcal{G}, \delta(\hat{g}) = \delta^* \wedge \mathcal{E}_m(\hat{g}) \geq k\epsilon \\
&\quad \implies (\exists c, \text{FN}_g(c) \geq \beta_1) \\
&\qquad \vee (\exists c_1, c_2, c', \min(\text{CO}_g(c_1, c'), \text{CO}_g(c_2, c')) \geq \beta_2) \wedge (FP_g \leq n\beta_1)
\end{aligned}$$

For

$$\beta_2 = \frac{\#^* k\epsilon}{2n(n-1)}$$

and

$$\beta_1 = \frac{\alpha\beta_2}{4\#_{\max}|\Delta|}$$

We proceed by contrapositive, assuming that $(\forall c, \text{FN}_g(c) < \beta_1)$ and $(\forall c_1, c_2, c', \min(\text{CO}_g(c_1, c'), \text{CO}_g(c_2, c')) < \beta_2) \vee (FP_g > n\beta_1)$ both hold. Note that this proof relies on none of our assumptions and is just about counting the outputs of $\hat{g}$.

Bounding CM and FP First, we bound CM and FP. Specifically, we establish that

$$\begin{aligned}
\sum_{c,c'} \text{CO}_g(c,c') &= \sum_c \mathbb{E}_{x,m^*}\left[\sum_{i \in \omega_c(m^*)} \sum_{c'} \mathbf{1}\left(v_{\hat{g}(x)}(i) = 1 \wedge v_{\hat{g}(x)}(i,c') = 1\right)\right] \\
&= \sum_c \mathbb{E}_{x,m^*}\left[\sum_{i \in \omega_c(m^*)} \mathbf{1}(v_{\hat{g}(x)}(i) = 1)\right] \\
\sum_{c,c'} \text{CO}_g(c,c') + 2\sum_c \text{CM}_g(c) &= \sum_c \mathbb{E}_{x,m^*}\left[\sum_{i \in \omega_c(m^*)} \mathbf{1}(v_{\hat{g}(x)}(i) = 1) + 2 \cdot \mathbf{1}\left(v_{\hat{g}(x)}(i) > 1\right)\right] \\
&\leq \sum_c \mathbb{E}_{x,m^*}\left[\sum_{i \in \omega_c(m^*)} v_{\hat{g}(x)}(i)\right] \\
&= \sum_c \mathbb{E}_{x,m^*}\left[\sum_{i \in I} \hat{g}(x)[i,c]\right] - \text{FP}_g \\
&= \#(\hat{g}) - \text{FP}_g \\
\text{FP}_g + \sum_{c,c'} \text{CO}_g(c,c') + 2\sum_c \text{CM}_g(c) &\leq \#* \\
\sum_c \text{FN}_g(c) + \sum_{c,c'} \text{CO}_g(c,c') + \sum_c \text{CM}_g(c) &= \sum_c \mathbb{E}_{x,m^*}\left[\sum_{i \in \omega_c(m^*)} 1\right] \\
&= \#^* \\
\sum_c \text{FN}_g(c) + \sum_{c,c'} \text{CO}_g(c,c') + \sum_c \text{CM}_g(c) &\geq \text{FP}_g + \sum_{c,c'} \text{CO}_g(c,c') + 2\sum_c \text{CM}_g(c) \\
\sum_c \text{FN}_g(c) &\geq \text{FP}_g + \sum_c \text{CM}_g(c)
\end{aligned}$$

And thus we have a bound on CM and FP in terms of FN. Note that this means we can eliminate the $\text{FP}_g > n\beta_1$ disjunction from our premises as we now know that $\text{FP}_g \leq \sum_c \text{FN}_g(c) \leq n\beta_1$.

Low FN implies high CO From above we have

$$\sum_c \text{FN}_g(c) + \sum_{c,c'} \text{CO}_g(c,c') + \sum_c \text{CM}_g(c) = \#^*$$

From this and the previous result it is clear that

$$2\sum_c \text{FN}_g(c) + \sum_{c,c'} \text{CO}_g(c,c') \geq \#^*$$

We then can state

$$\sum_{c,c'} \text{CO}_g(c,c') \geq \#^* - 2\sum_c \text{FN}_g(c)$$

High CO implies low $\mathcal{E}_m$ We now define the following function $\pi : [n] \to [n]$ assigning the "proper channel" of a given channel of $\hat{g}$ as

$$\pi(c') = \arg\max_c CO_g(c,c')$$

Assume that $\forall c_1, c_2, c', \min(\text{CO}_g(c_1,c'), \text{CO}_g(c_2,c')) \leq \beta_2$. We then have that

$$\forall c \neq \pi(c'), \text{CO}_g(c,c') \leq \min(\text{CO}_g(c,c'), \text{CO}_g(\pi(c'),c')) \leq \beta_2$$

Finally, we have that

$$\sum_{c,c'} \text{CO}_g(c,c') \leq n(n-1)\beta_2 + \sum_{c'} \text{CO}_g(\pi(c'),c')$$

We then express

$$\begin{aligned}
\sum_{c'} \text{CO}_g(\pi(c'),c') &\leq \sum_c \sum_{c'|\pi(c')=c} \text{CO}_g(c,c') \\
&= \sum_c \sum_{c'|\pi(c')=c} \mathbb{E}_{x,m^*}\left[\sum_{i\in\omega_c(m^*)} \mathbf{1}\left(v_{\hat{g}(x)}(i) = 1 \wedge v_{\hat{g}(x)}(i,c') = 1\right)\right] \\
&\leq \#^* - \#^*\mathcal{E}_m(\hat{g})
\end{aligned}$$

where the last step is viable as $\max(\#^*, \#(\hat{g})) = \#^*$ as $\delta(\hat{g}) = \delta^*$ We thus have that

$$\sum_{c,c'} \text{CO}_g(c,c') \leq n(n-1)\beta_2 + \#^* - \#^*\mathcal{E}_m(\hat{g})$$

and therefore

$$\#^*\mathcal{E}_m(\hat{g}) \leq n(n-1)\beta_2 + \#^* - \sum_{c,c'} \text{CO}_g(c,c')$$

Final proof We can then add the assumption $\forall c, \text{FN}(c) \leq \beta_1$. This means that

$$\sum_{c,c'} \text{CO}_g(c,c') \geq \#^* - 2n\beta_1$$

Putting this together with the above, we have

$$\#^*\mathcal{E}_m(\hat{g}) \leq n(n-1)\beta_2 + 2n\beta_1 \leq n(n-1)\beta_2 + n\beta_2 = (n(n-1)+n)\beta_2 \leq 2n(n-1)\beta_2 = \#^* k\epsilon$$

Thus ending our proof

D.2.2 COROLLARY: MOTIF ERROR AT ALL POSITIONS

We define the extended footprint of a cell as a function $\phi : I \to 2^{I\times[d]}$ mapping a location to the set of locations in the input whose output is in $p_2(i)$

$$\phi(i) = \{i'' : \exists i' \in p_2(i), i'' \in p(i')\}$$

Now, we establish that motif error in some percentage of positions implies a consistent probability of motif error every time the motif shows up, regardless of skeleton. First, define $\tilde{P}_{c,i}$ to be a distribution over regions of size $\phi(i)$ defined as

$$\tilde{P}_{c,i}(\eta) = P[x[\phi(i)] = \eta | g^*(x)[i, c] \neq 0]$$

We can use INPUT-FACTORIZATION to break this down as (letting o be the relative position of i within η

$$\begin{aligned}\tilde{P}_{c,i}(\eta) &= P[x[\phi(i)] = \eta | g^*(x)[i, c] \neq 0] \\ &= P_c[\eta[p(i) - o]] P_b[x[\phi(i) \setminus p(i)] = \eta[(\phi(i) \setminus p(i)) - o]]\end{aligned}$$

We implicitly use NON-OVERLAPPING when we assume that $\phi(i) \setminus p(i)$ is entirely over the background. The specific property here is that $\phi(i) \cap p(i') = \emptyset$ for all $i, i' \in m^*$, $i' \neq i$. This follows from NON-OVERLAPPING as we have that

$$\begin{aligned}\phi(i) \cap p(i') \neq \emptyset &\iff \exists i'', i'' \in \phi(i) \cap p(i') \\ &\iff \exists i'', i'' \in \phi(i) \wedge i'' \in p(i') \\ &\iff \exists i'', (\exists j \in p_2(i), i'' \in p(j)) \wedge i'' \in p(i') \\ &\iff \exists j, j \in p_2(i) \wedge \exists i'', i'' \in p(j) \wedge i'' \in p(i') \\ &\iff \exists j, j \in p_2(i) \wedge (p(j) \cap p(i')) \neq \emptyset \\ &\iff \exists j, j \in p_2(i) \wedge j \in p_2(i') \\ &\iff p_2(i) \cap p_2(i') \neq \emptyset\end{aligned}$$

Which is the exact condition given in NON-OVERLAPPING.

Note that this is no longer in any way dependent on i due to the translational invariance of P_b. Therefore, we have that $\tilde{P}_{c,i}(\eta) = \tilde{P}_c(\eta)$, and this is consistent at all locations that c appears, regardless of the skeleton.

We also define $q : 2^n \times \Delta \to 2^{\Delta \times [n]}$ be be a function that takes a vector u and offset d and returns the map $q(u, d)$ such that $q(u, d)[d] = u \wedge \forall d' \neq d, q(u, d)[d'] = 0$. Let $Q(u) = \{q(u, d) : d \in \Delta\}$

Claim The claim we wish to establish is

$$\begin{aligned}&\forall \hat{h} \in M \to Y, \hat{g} \in \mathcal{G}, \delta(\hat{g}) = \delta^* \wedge \mathcal{E}_m(\hat{g}) \geq k\epsilon \\ &\quad \implies \left(\exists c, P\left[\hat{g}(\eta) = 0 | \eta \sim \tilde{P}_c\right] \geq \frac{\beta_1}{\#_{\max}}\right) \\ &\qquad \vee (\\ &\qquad\quad \left(\exists c_1, c_2, c', \min_{c \in \{c_1, c_2\}} P\left[\hat{g}(\eta) \in Q(\mathbf{e}_{c'}) | \eta \sim \tilde{P}_c\right] \geq \frac{\beta_2}{\#_{\max}}\right) \\ &\qquad\quad \wedge \\ &\qquad\quad (FP_g \leq n\beta_1) \\ &\qquad)\end{aligned}$$

False negative case We now start with the assumption that $\text{FN}_{\hat{g}}(c) \geq \beta$. We have that

$$\begin{aligned}
\text{FN}_g(c) &= \mathbb{E}_{x,m^*}\left[\sum_{i \in \omega_c(m^*)} \mathbf{1}(v_{\hat{g}(x)}(i) = 0)\right] \\
&= \sum_m \mathbb{E}_{x,m^*}\left[\sum_{i \in \omega_c(m^*)} \mathbf{1}(v_{\hat{g}(x)}(i) = 0) | g^*(x) = m\right] P_m(m) \\
&= \sum_m \sum_{i \in m} \mathbb{E}_{x,m^*}\left[\mathbf{1}(v_{\hat{g}(x)}(i) = 0) | g^*(x) = m\right] P_m(m) \\
&= \sum_m \sum_{i \in m} P\left[\mathbf{1}(v_{\hat{g}(x)}(i) = 0) | g^*(x) = m\right] P_m(m) \\
&= \sum_m \sum_{i \in m} P\left[\hat{g}(\eta) = 0 | \eta \sim \tilde{P}_c\right] P_m(m) \\
&= P\left[\hat{g}(\eta) = 0 | \eta \sim \tilde{P}_c\right] \sum_m \sum_{i \in m} P_m(m) \\
&= P\left[\hat{g}(\eta) = 0 | \eta \sim \tilde{P}_c\right] \mathbb{E}\left[\sum_{i \in m} 1\right] \\
&= P\left[\hat{g}(\eta) = 0 | \eta \sim \tilde{P}_c\right] \#_c
\end{aligned}$$

and thus we can conclude that $P\left[\hat{g}(\eta) = 0 | \eta \sim \tilde{P}_c\right] \geq \frac{\beta_1}{\#_c} \geq \frac{\beta_1}{\#_{\max}}$.

Confusion Case

In this case, we have two properties, first that we have some c_1 and c_2 such that

$$\text{CO}_g(c_1, c') \geq \beta_2 \wedge \text{CO}_g(c_2, c') \geq \beta_2$$

and the second that

$$\text{FP}_g \leq \beta_1$$

First, we use a similar argument to the previous case to establish that

$$\begin{aligned}
\text{CO}_g(c, c') &= \mathbb{E}_{x,m^*}\left[\sum_{i \in \omega_c(m^*)} \mathbf{1}(v_{\hat{g}(x)}(i) = 1 \wedge v_{\hat{g}(x)}(i, c') = 1)\right] \\
&= \sum_m \mathbb{E}_{x,m^*}\left[\sum_{i \in \omega_c(m^*)} \mathbf{1}(v_{\hat{g}(x)}(i) = 1 \wedge v_{\hat{g}(x)}(i, c') = 1 | g^*(x) = m\right] P_m(m) \\
&= \sum_m \sum_{i \in m} \mathbb{E}_{x,m^*}\left[\mathbf{1}(v_{\hat{g}(x)}(i) = 1 \wedge v_{\hat{g}(x)}(i, c') = 1 | g^*(x) = m\right] P_m(m) \\
&= \sum_m \sum_{i \in m} P\left[\mathbf{1}(v_{\hat{g}(x)}(i) = 1 \wedge v_{\hat{g}(x)}(i, c') = 1) | g^*(x) = m\right] P_m(m) \\
&= \sum_m \sum_{i \in m} P\left[\hat{g}(\eta) \in Q(\mathbf{e}_{c'}) | \eta \sim \tilde{P}_c\right] P_m(m) \\
&= P\left[\hat{g}(\eta) \in Q(\mathbf{e}_{c'}) | \eta \sim \tilde{P}_c\right] \sum_m \sum_{i \in m} P_m(m) \\
&= P\left[\hat{g}(\eta) \in Q(\mathbf{e}_{c'}) | \eta \sim \tilde{P}_c\right] \mathbb{E}\left[\sum_{i \in m} 1\right] \\
&= P\left[\hat{g}(\eta) \in Q(\mathbf{e}_{c'}) | \eta \sim \tilde{P}_c\right] \#_c
\end{aligned}$$

and thus we can conclude that $P\left[\hat{g}(\eta) \in Q(\mathbf{e}_{c'}) | \eta \sim \tilde{P}_c\right] \geq \frac{\beta_2}{\#_c} \geq \frac{\beta_2}{\#_{\max}}$ for $c \in \{c_1, c_2\}$.

D.2.3 LEMMA: INDISTINGUISHABLE LOCAL-TO-GLOBAL

Statement: given a pairing scheme ψ, a predicate $\zeta : \mathbb{R}^{I\times[d]} \to \mathbb{B}$, some $\kappa > 0$, and that for all $\psi(m_2|m_1) > 0$ that are an i-OFF-BY-ONE PAIR and for all $x_R \in \mathbb{R}^{I\times[d]\setminus\phi(i)}$ we can assume

$$P[\zeta(x)|m_1, x_R] + P[\zeta(x)|m_2, x_R] \geq \kappa$$

we can prove that

$$P[\zeta(x)] \geq \frac{1}{2}\alpha\kappa$$

We begin by multiplying by $P[x_R|m_1] = P[x_R|m_2]$ (these are equal because m_1 and m_2 agree outside of $\phi(i)$)

$$\begin{aligned} P[\zeta(x)|m_1, x_R]P[x_R|m_1] + P[\zeta(x)|m_2, x_R]P[x_R|m_2] &\geq \kappa P[x_R|m_1] \\ P[\zeta(x), x_R|m_1] + P[\zeta(x), x_R|m_2] &\geq \kappa P[x_R|m_1] \end{aligned}$$

and integrating

$$\begin{aligned} \int P[\zeta(x), x_R|m_1]\mathrm{d}x_R + \int P[\zeta(x), x_R|m_2]\mathrm{d}x_R &\geq \kappa \int P[x_R|m_1]\mathrm{d}x_R \\ P[\zeta(x)|m_1] + P[\zeta(x)|m_2] &\geq \kappa \end{aligned}$$

We then multiply both sides by $\psi(m_2|m_1)P_m(m_1)$ and sum:

$$\sum_{m_1,m_2\in M} \psi(m_2|m_1)P_m(m_1)(P[\zeta(x)|m_1] + P[\zeta(x)|m_2]) \geq \sum_{m_1,m_2\in M} \psi(m_2|m_1)P_m(m_1)\kappa$$

We have that

$$\begin{aligned} \text{LHS} &= \sum_{m_1,m_2\in M} \psi(m_2|m_1)P_m(m_1)(P[\zeta(x)|m_1] + P[\zeta(x)|m_2]) \\ &= \sum_{m_1,m_2\in M} \psi(m_2|m_1)P_m(m_1)P[\zeta(x)|m_1] + \sum_{m_1,m_2\in M} \psi(m_2|m_1)P_m(m_1)P[\zeta(x)|m_2] \\ &= \sum_{m_2\in M} \psi(m_2|m_1)P[\zeta(x)] + \sum_{m_2\in M} q(m_2)P[\zeta(x)|m_2] \\ &\leq P[\zeta(x)] + \sum_{m_2\in M} P_m(m_2)P[\zeta(x)|m_2] \\ &\leq P[\zeta(x)] + P[\zeta(x)] \\ P[\zeta(x)] &\geq \frac{1}{2}\text{LHS} \\ \text{RHS} &= \sum_{m_1,m_2\in M} \psi(m_2|m_1)P_m(m_1)\kappa \\ &= \sum_{m_2\in M} q(m_2)\kappa \\ &\geq \alpha\kappa \end{aligned}$$

and therefore, we have that

$$P[\zeta(x)] \geq \frac{1}{2}\kappa\alpha$$

which completes our proof.

D.2.4 LEMMA: FALSE NEGATIVES

Given some $\hat{g}$, $\kappa < \frac{1}{2}$ such that

$$P[\hat{g}(\eta_1) = 0|\eta_1 \sim \tilde{P}_c] \geq \kappa$$

we have that for all $\hat{h}$,

$$\mathcal{E}(\hat{h} \circ \hat{g}) \geq \frac{1}{2}\alpha\kappa$$

We now proceed with our proof. Let ψ be the pairing scheme corresponding to $v_1 = \mathbf{e}_c$ and $v_2 = \mathbf{0}$, and $d_1 = d_2 = 0$. Fix any m_1, m_2, i such that $\psi(m_2|m_1) > 0$ being an i-OFF-BY-ONE PAIR and $x_R \in \mathbb{R}^{I \times [d] \setminus \phi(i)}$. We can now see that

$$\begin{aligned}
P[\hat{f}(x) \neq y^*|m_1, x_R] &\geq P[\hat{f}(x) \neq y^*, \hat{g}(\phi(i)) = 0|m_1, x_R] \\
&= P[\hat{f}(x) \neq y^*|\hat{g}(\phi(i)) = 0, m_1, x_R] P[\hat{g}(\phi(i)) = 0|m_1, x_R] \\
&= P[\hat{f}(x) \neq y^*|\hat{g}(\phi(i)) = 0, m_1, x_R] P[\hat{g}(\eta_1) = 0|\eta_1 \sim \tilde{P}_c] \\
&\geq P[\hat{f}(x) \neq y^*|\hat{g}(\phi(i)) = 0, m_1, x_R]\kappa
\end{aligned}$$

Once we know x_R and that $\hat{g}(\phi(i)) = 0$, we know that $\hat{f}(x)$ is entirely dependent on x_R and not on $x[p(i)]$ because we have access via $\hat{g}(\phi(i))$ to all values of $\hat{g}(x)$ that are influenced by $x[p(i)]$. As such, we can replace $\hat{f}(x)$ with $\lambda(x_R)$. Thus, we have

$$P[\hat{f}(x) \neq y^*|m_1, x_R] \geq P[\lambda(x_R) \neq y^*|m_1, x_R]\kappa$$

By the translational property of P_b we know that $P[\hat{g}(\eta_1) = 0|\eta_1 \sim P_b]$ is a definable, fixed, quantity, and applies to any random variable $\eta_i = x[\phi(i)]$. We thus have that $P[\hat{g}(\eta_1) = 0|\eta_1 \sim P_b] \geq 1 - n\delta$ as otherwise $\hat{g}$ would not be capable of having a density of δ on the whole image. We can safely assume $n\delta < \frac{1}{2}$ since otherwise the NON-OVERLAPPING would be violated, since the minimum size of a motif cell is 3 (1-dimensional, radius 1). Thus we can assume that

$$P[\hat{g}(\eta_1) = 0|\eta_1 \sim P_b] \geq \frac{1}{2} \geq \kappa$$

and thus have the same property

$$P[\hat{f}(x) \neq y^*|m_2, x_R] \geq P[\lambda(x_R) \neq y^*|m_2, x_R]\kappa$$

We have that $h^*(m_1) \neq h^*(m_2)$. We can then proceed

$$\begin{aligned}
P[\hat{f}(x) \neq y^*|m_1, x_R] + P[\hat{f}(x) \neq y^*|m_2, x_R] &\geq \kappa(P[\lambda(x_R) \neq y^*|m_1, x_R] + P[\lambda(x_R) \neq y^*|m_2, x_R]) \\
&\geq \kappa(P[\lambda(x_R) \neq h^*(m_1)|m_1, x_R] + P[\lambda(x_R) \neq h^*(m_2)|m_2, x_R]) \\
&= \kappa(\mathbf{1}(\lambda(x_R) \neq h^*(m_1)) + \mathbf{1}(\lambda(x_R) \neq h^*(m_2)) \\
&\geq \kappa(\mathbf{1}(\lambda(x_R) \neq h^*(m_1) \vee \lambda(x_R) \neq h^*(m_2)) \\
&= \kappa
\end{aligned}$$

As such, we can now apply Lemma D.2.3 to get the statement

$$P[\hat{f}(x) \neq y^*] \geq \frac{1}{2}\kappa\alpha$$

which completes our proof

D.2.5 LEMMA: CONFUSION

Given c_1, c_2, and c', κ, and some $\hat{g}$ such that

$$P[\hat{g}(\eta_1) \in Q(\mathbf{e}_{c'})|\eta_1 \sim \tilde{P}_{c_1}] \geq \kappa \wedge P[\hat{g}(\eta_2) \in Q(\mathbf{e}_{c'})|\eta_2 \sim \tilde{P}_{c_2}] \geq \kappa$$

we have that for all $\hat{h}$

$$\mathbb{E}[\hat{f}(x) \neq y^* \vee \text{FP}_g(x) > 0] \geq \frac{\alpha\kappa}{2|\Delta|}$$

We now proceed with our proof.

We proceed as in the proof of Lemma D.2.4, with a few variations. Consider the pairing scheme ψ corresponding to $v_1 = \mathbf{e}_{c_1}$ and $v_2 = \mathbf{e}_{c_2}$, and $d_1 = -\arg\max_d P[\hat{g}(\eta_1) = q(u, d)|\eta_1 \sim \tilde{P}_{v_1}]$ and $d_2 = -\arg\max_d P[\hat{g}(\eta_2) = q(u, d)|\eta_2 \sim \tilde{P}_{v_2}]$. We have that $P[\hat{g}(\eta_1) = q(u, -d_1)|\eta_1 \sim \tilde{P}_{v_1}], P[\hat{g}(\eta_2) = q(u, -d_2)|\eta_2 \sim \tilde{P}_{v_2}] \geq \kappa/|\Delta|$. Let $\kappa' = \kappa/|\Delta|$

Let (m_1, m_2) be an i-OFF-BY-ONE PAIR such that $\psi(m_2|m_1) > 0$. Fix $x_R \in \mathbb{R}^{I\times[d]\setminus\phi(i)}$. Consider

$$P[\hat{f}(x) \neq y^* \vee \text{FP}_g(x) > 0|m_1, x_R] + P[\hat{f}(x) \neq y^* \vee \text{FP}_g(x) > 0|m_2, x_R]$$

We have that $h^*(m_1) \neq h^*(m_2)$. We also know that

$$P[\hat{f}(x) \neq y^* \vee \text{FP}_g(x) > 0|m_1, x_R] \geq P[\hat{f}(x) \neq h^*(m_1) \vee \text{FP}_g(x) > 0|m_1, x_R]$$

We can then analyze

$$\begin{aligned}
P[\hat{f}(x) \neq h^*(m_1) \vee \text{FP}_g(x) > 0|m_1, x_R] &\geq P[\hat{f}(x) \neq h^*(m_1) \vee \text{FP}_g(x) > 0, \hat{g}(x[\phi(i-d_1)]) = q(u,0)|m_1, x_R] \\
&= P[\hat{f}(x) \neq h^*(m_1) \vee \text{FP}_g(x) > 0|m_1, x_R, x[\phi(i-d_1)] = q(u,0)]P[x[\phi(i-d_1)] \\
&\geq P[\hat{f}(x) \neq h^*(m_1) \vee \text{FP}_g(x) > 0|m_1, x_R, \hat{g}(x[\phi(i-d_1)]) = q(u,0)]\kappa'
\end{aligned}$$

where the last step comes from the fact that m_1 has its motif at $i + d_1$, and therefore, $\hat{g}$ should activate at i. Finally, if we define $\lambda(x_R)$ to be the value $\hat{h}(\hat{m})$ takes when $\hat{m}[i'] = \hat{g}(x[\phi(i')])$ for all i' in a motif cell of m^* other than i and 0 otherwise, we have that

$$\hat{f}(x) \neq \lambda(x_R) \implies \text{FP}_g(x) > 0$$

because if it is equal to any other value, that indicates that $\hat{g}$ is sending some values through non-motif cell channels. We thus have that

$$\hat{f}(x) \neq h^*(m_1) \vee \text{FP}_g(x) > 0 \iff \lambda(x_R) \neq h^*(m_1) \vee \text{FP}_g(x) > 0$$

Thus, we have that

$$P[\hat{f}(x) \neq h^*(m_1) \vee \text{FP}_g(x) > 0|m_1, x_R] \geq \kappa' P[\lambda(x_R) \neq h^*(m_1) \vee \text{FP}_g(x) > 0|m_1, x_R]$$

and by an identical argument

$$P[\hat{f}(x) \neq h^*(m_2) \vee \text{FP}_g(x) > 0|m_2, x_R] \geq \kappa' P[\lambda(x_R) \neq h^*(m_2) \vee \text{FP}_g(x) > 0|m_2, x_R]$$

We now proceed by cases. Either $\lambda(x_R) \neq h^*(m_1)$, in which case

$$P[\lambda(x_R) \neq h^*(m_1) \vee \text{FP}_g(x) > 0|m_1, x_R] = 1$$

or $\hat{f}(x) = h^*(m_1)$ and thus $\lambda(x_R) \neq h^*(m_2)$ and thus

$$P[\lambda(x_R) \neq h^*(m_2) \vee \text{FP}_g(x) > 0|m_2, x_R] = 1$$

In either case, we have

$$P[\hat{f}(x) \neq y^* \vee \text{FP}_g(x) > 0|m_1, x_R] + P[\hat{f}(x) \neq y^* \vee \text{FP}_g(x) > 0|m_2, x_R] \geq \kappa'$$

Applying Lemma D.2.3 to this statement, we get that we have

$$P[\hat{f}(x) \neq y^* \vee \text{FP}_g(x) > 0] \geq \frac{1}{2}\kappa'\alpha$$

completing our proof

D.3 MAIN PROOF

The statement is reproduced below:

$$\forall \hat{g} \in \mathcal{G} \,.\, \delta(\hat{g}) = \delta^* \implies \left(\forall \epsilon > 0, \mathcal{E}(\hat{h} \circ \hat{g}) < \epsilon \implies \mathcal{E}_m(\hat{g}) < k\epsilon\right)$$

Let

$$k = \frac{16\#^2_{\max}|\Delta|n(n-1)}{\#^*\alpha^2}$$

and then fix $\hat{g}$ such that $\delta(\hat{g}) = \delta^*$ and $\epsilon > 0$. Assume towards contradiction that the statement $\mathcal{E}(\hat{h} \circ \hat{g}) < \epsilon \implies \mathcal{E}_m(\hat{g}) < k\epsilon$ is false. We then have $\mathcal{E}(\hat{h} \circ \hat{g}) < \epsilon$ and $\mathcal{E}_m(\hat{g}) \geq k\epsilon$. Using Corrolary D.2.2 we have two cases.

D.3.1 FALSE NEGATIVE CASE

We have that there is some c for which

$$P\left[\hat{g}(\eta) = 0 | \eta \sim \tilde{P}_c\right] \geq \frac{\beta_1}{\#_{\max}}$$

Applying Lemma D.2.4, we have that

$$\mathcal{E}(\hat{h} \circ \hat{g}) \geq \frac{1}{2}\alpha\frac{\beta_1}{\#_{\max}} = \frac{\alpha^2\beta_2}{8\#_{\max}^2|\Delta|} = \frac{\alpha^2\#^* k\epsilon}{16\#_{\max}^2|\Delta|n(n-1)} = \epsilon$$

which is a contradiction with $\mathcal{E}(\hat{h} \circ \hat{g}) < \epsilon$.

D.3.2 CONFUSION CASE

We have that there exist some c_1, c_2, c' such that

$$\min_{c \in \{c_1, c_2\}} P\left[\hat{g}(\eta) \in Q(\mathbf{e}_{c'}) | \eta \sim \tilde{P}_c\right] \geq \frac{\beta_2}{\#_{\max}}$$

and also,

$$FP_g \leq n\beta_1$$

Applying Lemma D.2.5, we have that

$$\mathbb{E}[\hat{f}(x) \neq y^* \vee \text{FP}_g(x)] \geq \frac{1}{2}\alpha\frac{\beta_2}{|\Delta|\#_{\max}}$$

We also know that

$$\mathbb{E}[\text{FP}_g(x)] \leq \beta_1$$

and therefore

$$\mathcal{E}(\hat{f}) \geq \frac{1}{2}\alpha\frac{\beta_2}{|\Delta|\#_{\max}} - \beta_1 = \frac{1}{4}\alpha\frac{\beta_2}{|\Delta|\#_{\max}} = \frac{1}{8}\alpha\frac{\#^* k\epsilon}{n(n-1)|\Delta|\#_{\max}} = \frac{2\#_{\max}\epsilon}{\alpha} > \epsilon$$

which is a contradiction with $\mathcal{E}(\hat{h} \circ \hat{g}) < \epsilon$, thus completing our proof.

E MOTIF ERROR EQUIVALENCE

In this section, we prove that our proof's error metric is only ever off by a constant factor from our empirical error metrics.

E.1 FORMAL STATEMENT

For all $\hat{g} \in \mathcal{G}$ such that $\delta(\hat{g}) = \delta^*$,

$$\begin{aligned}
\mathcal{E}_m(\hat{g}) \leq \epsilon &\implies \text{FNE}(\hat{g}) \leq \epsilon \wedge \text{FPE}(\hat{g}) \leq \epsilon \wedge \text{CE}(\hat{g}) \leq 2\epsilon \\
\mathcal{E}_m(\hat{g}) \leq \epsilon &\impliedby \text{FNE}(\hat{g}) \leq \frac{1}{4}\epsilon \wedge \text{CE}(\hat{g}) \leq \frac{1}{2}\epsilon
\end{aligned}$$

E.2 CORRESPONDENCE WITH QUANTITIES FROM LEMMA D.2.1

First, note that

$$\#^*\mathcal{E}_m(\hat{g}) = \min_{\tau} \#^* - \sum_{c} \sum_{c' | c = \tau(c')} \text{CO}_g(c, c')$$

Then, note that

$$\text{FNE}(\hat{g}) = \frac{\sum_c \text{FN}_g(c)}{\#^*}$$
$$\text{FPE}(\hat{g}) = \frac{\text{FP}_g}{\#(\hat{g})}$$

by inspection. The case of CE is more complicated, due to the presence of MM. Inspecting the denominator, we have

$$|\text{MM}(\hat{m}, m^*)| + |\text{NMM}(\hat{m}, m^*)| = |P(\hat{m})| - |\text{FPM}(\hat{m}, m^*)|$$

and therefore

$$\mathbb{E}[|\text{MM}(\hat{g}(x), g^*(x))|] + \mathbb{E}[|\text{NMM}(\hat{g}(x), g^*(x))|] = \#(\hat{g})(1 - \text{FPE}(\hat{g}))$$

Additionally, we can note that if we assume there are no ties in the $\max$ computation (or alternatively, they are broken in some systematic way rather than leading to duplicates), we know that

$$|\text{MM}(\hat{m}, m^*)| = |\{(i, c) \in P(g^*(x)) : \exists (i', c') \in P(\hat{g}(x)) : i' \in p_2(i)\}|$$

and thus

$$\mathbb{E}[|\text{MM}(\hat{g}(x), g^*(x))|] = \#^*(1 - \text{FNE}(\hat{g}))$$

Letting $Q(\hat{m}, m^*) = \{(i, c) \in P(g^*(x)) : \exists (i', c') \in P(\hat{g}(x)) : i' \in p_2(i)\}$ we can break this down into a dichotomy

$$Q(\hat{m}, m^*) = Q_1(\hat{m}, m^*) \sqcup Q_2(\hat{m}, m^*)$$

where

$$Q_1(\hat{m}, m^*) = \{(i, c) \in P(g^*(x)) : \exists! (i', c') \in P(\hat{g}(x)) : i' \in p_2(i)\}$$
$$Q_2(\hat{m}, m^*) = \{(i, c) \in P(g^*(x)) : \exists (i'_1, c'_1) \neq (i'_2, c'_2) \in P(\hat{g}(x)) : i'_1, i'_2 \in p_2(i)\}$$

We have that

$$\mathbb{E}[|Q_2(\hat{g}(x), g^*(x))|] = \sum_c \text{CM}_g(c)$$

Finally, we can see that

$$\begin{aligned}
|\text{conf}_\tau(\hat{m}, m^*)| &= \sum_{(i,c) \in Q} |\{(i', c') \in \text{conf}_\tau(\hat{m}, m^*) : i' \in p_2(i)\}| \\
&= \lambda_\tau(\hat{m}, m^*)|Q_2(\hat{m}, m^*)| + \sum_{(i,c) \in Q_1} |\{(i', c') \in \text{conf}_\tau(\hat{m}, m^*) : i' \in p_2(i)\}| \\
&= \lambda_\tau(\hat{m}, m^*)|Q_2(\hat{m}, m^*)| + \sum_{(i,c) \in Q_1} \mathbf{1}(\exists c', \tau(c') = c \wedge v_m(i) = 1 \wedge v_m(i, c') \neq 1) \\
&= \lambda_\tau(\hat{m}, m^*)|Q_2(\hat{m}, m^*)| + \sum_{(i,c) \in Q_1} \sum_{c' | \tau(c') = c} \mathbf{1}(v_m(i) = 1 \wedge v_m(i, c') = 1) \\
\mathbb{E}[|\text{conf}_\tau(\hat{g}(x), g^*(x))|] &= \lambda_\tau \sum_c \text{CM}_g(c) + \sum_c \sum_{c' | \tau(c') = c} \text{CO}_g(c, c') \\
&= \lambda_\tau \sum_c \text{CM}_g(c) + \sum_{c,c'} \text{CO}_g(c, c') - \sum_c \sum_{c' | \tau(c') = c} \text{CO}_g(c, c') \\
&= \lambda_\tau \sum_c \text{CM}_g(c) + \#^* - \sum_c \text{FN}_g(c) - \sum_c \text{CM}_g(c) - \sum_c \sum_{c' | \tau(c') = c} \text{CO}_g(c, c')
\end{aligned}$$

where $\lambda_\tau(\hat{m}, m^*)$ and λ_τ are some constants in $[0, 1]$.

Since $(\min_x A(x)) + (\min_x B(x)) \leq \min_x(A(x) + B(x)) \leq (\min_x A(x)) + (\max_x B(x))$, we have that

$$\min_\tau \mathbb{E}[|\text{conf}_\tau(\hat{g}(x), g^*(x))|] = \#^* \mathcal{E}_m(\hat{g}) + \lambda_1 \sum_c \text{CM}_g(c) - \sum_c \text{FN}_g(c) - \sum_c \text{CM}_g(c)$$

for some $\lambda_1 \in [0, 1]$ and thus

$$\begin{aligned}\min_\tau \mathbb{E}[|\text{conf}_\tau(\hat{g}(x), g^*(x))|] &= \#^* \mathcal{E}_m(\hat{g}) - \lambda_2 \sum_c \text{CM}_g(c) - \sum_c \text{FN}_g(c) \\ &= \#^* \mathcal{E}_m(\hat{g}) - \lambda_3 \sum_c \text{FN}_g(c) - \sum_c \text{FN}_g(c) \\ &= \#^* \mathcal{E}_m(\hat{g}) - (1 + \lambda_3)\text{FNE}(\hat{g})\#^*\end{aligned}$$

for some $\lambda_2 \in [0, 1]$, and since $\lambda_3 = \lambda_2 \frac{\sum_c \text{CM}_g(c)}{\sum_c \text{FN}_g(c)} \leq \lambda_2$, we have $\lambda_3 \in [0, 1]$. We thus have that

$$\begin{aligned}\text{CE}(\hat{g}) &= \#^* \frac{\mathcal{E}_m(\hat{g}) - (1 + \lambda_3)\text{FNE}(\hat{g})\#^*}{\#^*(1 - \text{FNE}(\hat{g}))} \\ &= \frac{\mathcal{E}_m(\hat{g}) - (1 + \lambda_3)\text{FNE}(\hat{g})}{1 - \text{FNE}(\hat{g})}\end{aligned}$$

E.3 Main Proof

Forward direction Assume $\mathcal{E}_m(\hat{g}) \leq \epsilon$. We have that

- We proceed by using the quantities from above, bounding FNE.

$$\begin{aligned}\text{FNE}(\hat{g}) &= \frac{\sum_c \text{FN}_g(c)}{\#^*} \\ &= \frac{\#^* \mathcal{E}_m(\hat{g}) - \lambda_2 \sum_c \text{CM}_g(c) - \min_\tau \mathbb{E}[|\text{conf}_\tau(\hat{g}(x), g^*(x))|]}{\#^*} \\ &\leq \mathcal{E}_m(\hat{g}) \\ &\leq \epsilon\end{aligned}$$

- Since we know from Section D.2.1 that $\text{FP}_g \leq \sum_c \text{FN}_g(c)$ we have

$$\begin{aligned}\text{FPE}(\hat{g}) &= \frac{\text{FP}_g}{\#^*} \\ &\leq \frac{\sum_c \text{FN}_g(c)}{\#^*} \\ &\leq \epsilon\end{aligned}$$

- If $\epsilon \geq \frac{1}{2}$ then clearly $\text{CE}(\hat{g}) \leq 1 = 2\epsilon$. If $\epsilon < \frac{1}{2}$ we have that $1 - \text{FNE}(\hat{g}) > \frac{1}{2}$ and thus

$$\begin{aligned}\text{CE}(\hat{g}) &= \frac{\mathcal{E}_m(\hat{g}) - (1 + \lambda_3)\text{FNE}(\hat{g})}{1 - \text{FNE}(\hat{g})} \\ &\leq 2(\mathcal{E}_m(\hat{g}) - (1 + \lambda_3)\text{FNE}(\hat{g})) \\ &\leq 2\mathcal{E}_m(\hat{g}) \\ &\leq 2\epsilon\end{aligned}$$

  as desired

Backward Direction Assume that $\text{CE}(\hat{g}) \leq \frac{1}{2}\epsilon$ and $\text{FNE}(\hat{g}) \leq \frac{1}{4}\epsilon$. We then have that

$$\begin{aligned}\mathcal{E}_m &= \text{CE}(\hat{g})(1 - \text{FNE}(\hat{g})) + (1 + \lambda_3)\text{FNE}(\hat{g}) \\ &\leq \text{CE}(\hat{g}) + 2\text{FNE}(\hat{g}) \\ &\leq \epsilon\end{aligned}$$

F Counterexamples for Less Intuitive Assumptions

F.1 Translational invariance of background distribution

We assume that P_b is translationally invariant. For an example of what happens if this assumption is broken, consider a version of DigitCircle where one digit always appears on the left side of

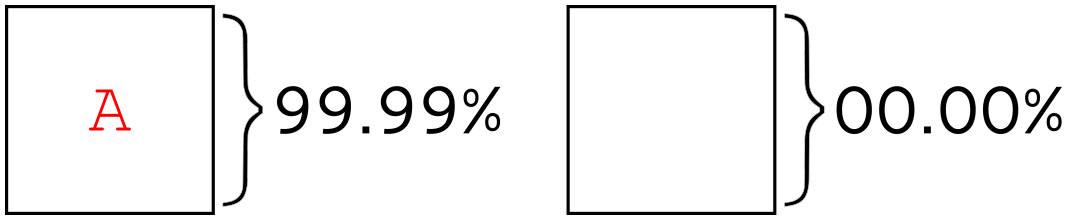

Figure 6: Counterexample motivating the $q(m_2) \leq P_m(m_2)$ requirement

the image, and is not read as part of the output y^*. While one might think this would lead to the possibility of high motif error without high end-to-end error, the locality of $\hat{g}$ ensures that the motif is predicted correctly despite not being used, as $\hat{g}$ does not "know" that the motif will not be useful. However, if P_b were not translationally invariant, it would be possible for e.g., the background to be systematically darker on the left side of the image, with the motif prediction being slightly off center to take this into account and not report the motif if it is in the darker region. This would not affect end-to-end error but would affect motif error.

F.2 No increase in probability mass for perturbations

To demonstrate the necessity of the $q(m_2) \leq P_m(m_2)$ requirement, consider the domain shown in Figure 6, which has exactly $M = \{m_1, m_2\}$ with m_1 having $1 - \iota$ probability (depicted is $\iota = 0.01\%$). Clearly, this domain trivially satisfies Non-Overlapping and Input-Factorization as there is only one motif. Additionally, if we let $\psi(m_2|m_1) = 1$ and $\psi(\perp|m_2) = 1$, we have that ψ clearly satisfies the support requirement and since $q(m_2) = 1 - \iota$, we have that $\sum_m q(m) = 1 - \iota$ so this domain satisfies the α-Motif-Importance assumption with $\alpha = 1 - \iota$. However, we have that we can set $\hat{g}(x) = 0$ and $\hat{h}(x) =$ `"A"`, giving $\mathcal{E}(\hat{h} \circ \hat{g}) = \iota$ and $\mathcal{E}_m(\hat{g}) = 1$. This clearly breaks our proof since we can make ι arbitrarily small while not changing α much as it converges to 1.

G Confusion

Figure 7 depicts appropriately permuted confusion matrices for each domain. Our model generally assigns each true motif to a channel or set of channels in the sparse layer. The main exception is that in LaTeX-OCR, the fraction bar is never recognized, and `()` are only sometimes recognized. In other seeds, + exhibits similar behavior to `()`.

H Sparsity as an Information Bound

H.1 Connection to Information Bound

Sparsity induces an information bound by limiting the amount of information in the intermediate representation. Specifically, if we let $\mathcal{X}$ be a random variable for the input, and $\mathcal{M} = g^*(\mathcal{X})$ be the motif layer, we have that we can bound the mutual information between inputs and motifs as $I(\mathcal{X}, \mathcal{M}) \leq H(\mathcal{M})$, where $H(\cdot)$ is entropy. Thus, to bound mutual information, it is sufficient to bound $H(\mathcal{M})$. We first can break it into per-channel components: $H(\mathcal{M}) \leq \sum_{i,c} H(\mathcal{M}[i, c])$,Then, let $\delta_{i,c}$ denote the density of channel c at position i, and $\eta \geq H(\mathcal{M}[i, c]|\mathcal{M}[i, c] \neq 0)$ be a bound on the amount of entropy in each nonzero activation (see Appendix H.2). Then we apply the chain rule to get $H(\mathcal{M}[i, c]) \leq H(B(\delta_{i,c})) + \eta\delta_{i,c}$ where $B(\cdot)$ is the Bernoulli distribution. Thus, $H(\mathcal{M}) \leq \sum_{i,c} H(B(\delta_{i,c})) + Sn\eta\delta$,where S is the size of the image in pixels and n is the number of channels, and δ is defined as in section 2. Finally, using Jensen's inequality (as $H(B(t))$ is concave):

$$I(\mathcal{X}, \mathcal{M}) \leq H(\mathcal{M}) \leq Sn(H(B(\delta)) + \eta\delta).$$

Since the computed bound is a monotonic function in δ, where as $\delta \to 0$, the bound approaches 0, we can see that a sparsity bound can be used as an informational bottleneck for any information bound of a user's choosing.

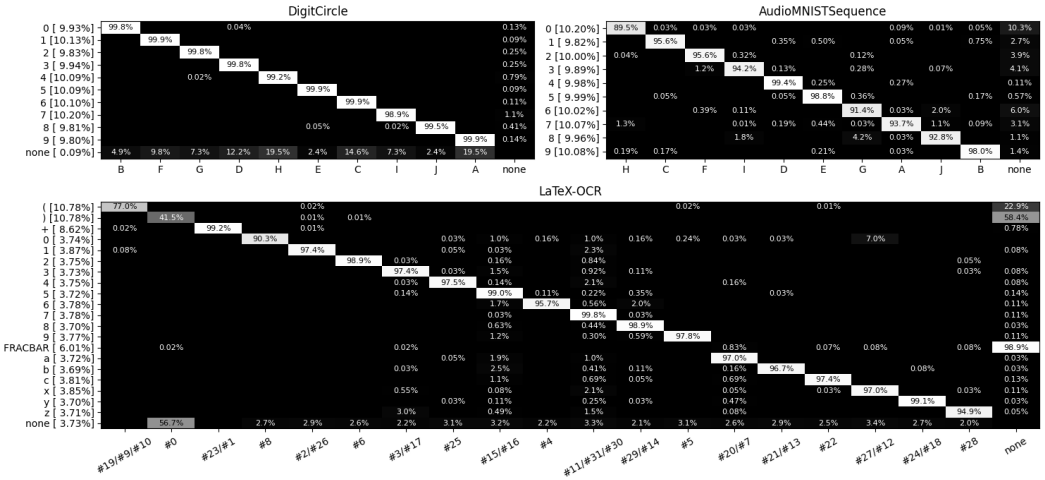

Figure 7: Confusion Matrix of 10k unseen samples computed for seed=1 across all domains. Each row represents a true motif being recognized and column represents a channel in the model's motif output. False positive and false negative motifs are placed into the `none` rows and columns, respectively. Each row is labeled by the percentage of motifs falling into the row, and each row's cells are then normalized to add to 1. We then permute to align along the diagonal. For LaTeX-OCR, we use more channels than there are symbol types so we merge channels together for display and analysis.

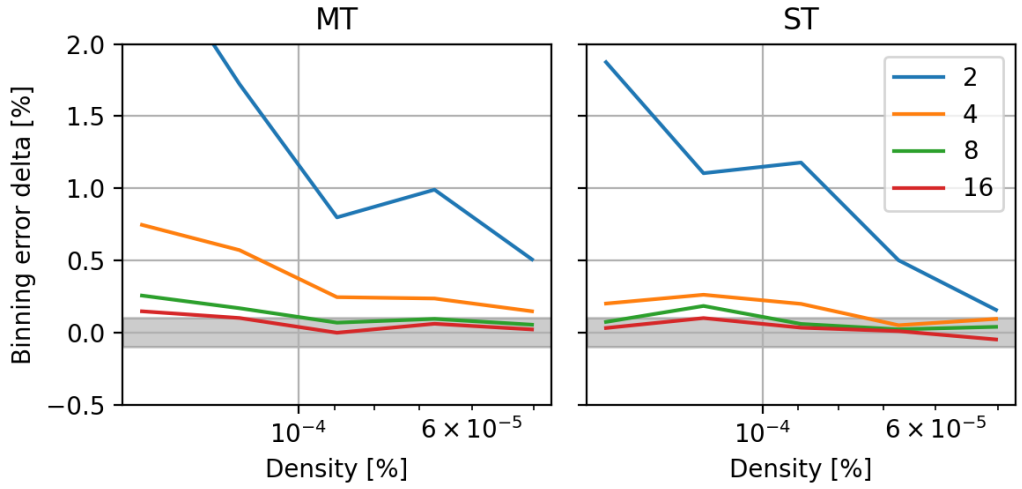

Figure 8: Increase in error when binning. Each series represents a different bin count, as annotated in the legend. Density is log-scaled and reversed to indicate training progress. MT is the model tracked in the rest of the paper, ST is the model as defined in Appendix J.3

H.2 Entropy upper bound

To compute our entropy upper bound, we must first compute η, as defined in Section H.1. To compute this, we bin the nonzero activations into 2^k bins by quantile. We set η to be the smallest value of k that does not substantially affect the accuracy of the model (we consider 0.1% to be a reasonable threshold for this purpose). Figure 8 shows the result of this experiment, averaged across 9 seeds. The general downward trend in error caused by binning as density decreases demonstrates

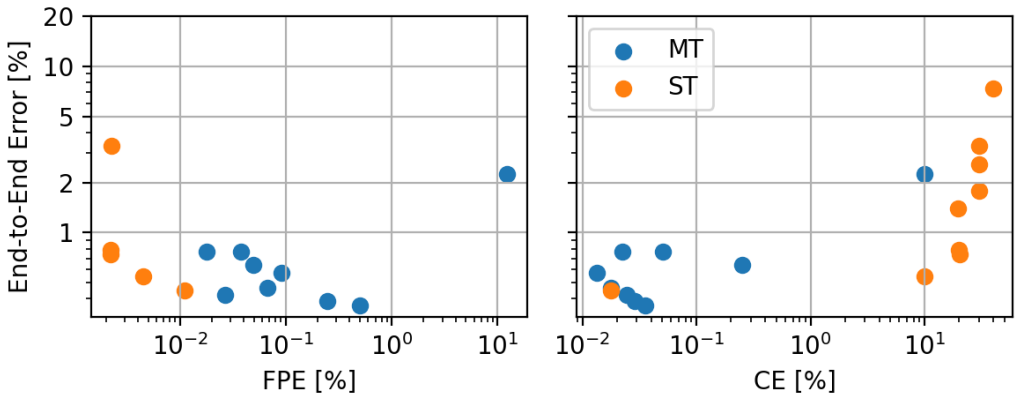

Figure 9: Model error versus FPE and CE, at 1.1× the minimum sparsity. All are log-scaled to highlight the low-error region. Each dot represents a single model training seed.

that reducing the number of motifs reduces the importance of the precise magnitudes. For the purposes of entropy bounding, we can use $\eta = \log(16) = 4$b.

I PREDICTING MOTIF ERROR.

Figure 9 shows the relationship between the motif errors and the overall end-to-end error for DIGIT-CIRCLE. There is no relationship for FPE, but there is a positive relationship for CE, implying that a strategy where one trains several models and then chooses the one with the best validation error is a good way to reduce CE and thereby improve motif quality. This provides further evidence for Motif Identifiability (though the primary evidence for this remains that this model is able to achieve low FPE and CE in general, as training itself focuses on reducing end-to-end error via the loss function). While this may seem to contradict the result in Section 5.2, it in fact does not. Within a single model, tightening the density has inverse effects on end-to-end error and CE, but separately, some models are in general more or less accurate.

J COMPARISONS BETWEEN SPARLING AND OTHER TECHNIQUES FOR SPARSITY

In this section we compare to alternatives of the SPARLING model. For all comparisons, we keep the model architecture fixed and only modify the Sparse layer.

J.1 BASELINES

We consider two other approaches to ensuring the creation of sparse motifs, both taking the form of auxiliary regularization losses. In both cases, we vary loss weight to analyze how that affects error and sparsity. First, we consider L_1 loss. In our implementation, we use an affine batch normalization layer followed by a ReLU. The output of the ReLU is then used in an auxiliary L_1 loss[9]. This approach is discussed in Jiang et al. (2015). We also consider using KL-divergence loss as in Jiang et al. (2015). The approach is to apply a sigmoid, then compute a KL-divergence between the Bernoulli implied by the mean activation of the sigmoid and a target sparsity value (we use 99.995% to perform a direct comparison). While this usually is done across the training data Ng (2011), we instead enforce the loss across all positions and channels, but per-batch (the mean sparsity should be similar in each batch). Our other modification, in order to induce true sparsity, is to, after the sigmoid layer (where the loss is computed), subtract 0.5 and apply a ReLU layer.

[9]This approach parameterizes the same model class as SPARLING; both act as a ReLU in a forward pass

	L_1 $\lambda = 0.1$	$\lambda = 1$	$\lambda = 2$	$\lambda = 5$	$\lambda = 10$	SPARLING MT
FPE [%]	99.99	99.90	91.25	95.99	97.63	1.48 [0.07-4.23]
FNE [%]	0.00	0.00	58.09	73.12	84.51	0.42 [0.25-0.67]
CE [%]	50.34	47.84	45.65	50.85	33.82	1.16 [0.03-3.39]
E2EE [%]	0.68	2.85	70.31	75.00	73.20	0.74 [0.47-1.15]
Density [%]	37	4.7	0.023	0.032	0.028	0.005

Table 1: Results of L_1 experiment on DIGITCIRCLE. As L_1 increases, the density decreases, but end-to-end error becomes $> 50\%$, and CE/FPE never improve to the level of SPARLING. SPARLING is able to keep error low while achieving lower density than L_1 with any λ value we tried.

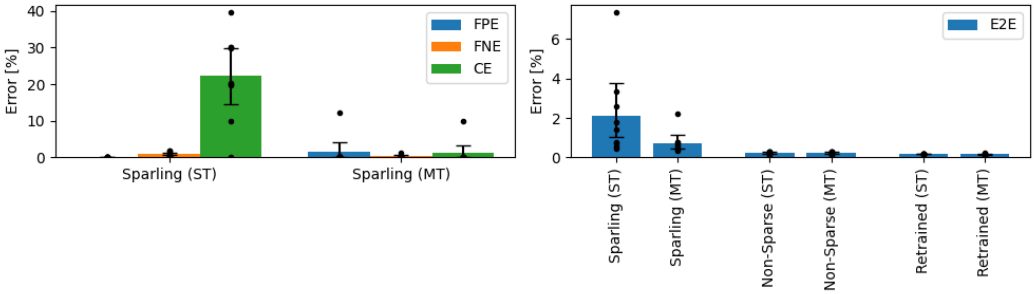

Figure 10: SPARLING using MT (as in the main figures) vs ST

Table 1 shows the results of using L_1 as a method for encouraging sparsity. There are two weight regimes, where when $\lambda \leq 1$, we end up with high density (relative to the theoretical minimum) but low error, and when $\lambda \geq 2$, we end up with high-error model. Even in the latter case, the L_1 loss does not consistently push density down to the level of SPARLING, suggesting it might be insufficiently strong as a learning signal. In our experiments, the KL-divergence was unable to achieve a density below 0.1%, even when we used a loss weight as high as $\lambda = 10^5$ and 3×10^6 steps (much more than was necessary for convergence of the L_1 model). Thus, we conclude that it is unsuitable for encouraging the kind of sparsity we are interested in.

J.2 ABLATIONS

We consider two ablations: First, is the batch normalization we place before our sparse layer necessary? Second, is the adaptive sparsity algorithm we use necessary? These ablations are only evaluated on DIGITCIRCLE as it is the domain where simpler techniques would work best.

We find that including a batch normalization before the sparsity layer is crucial. Without a batch normalization layer, over 9 runs, the best model gets an E2EE of 71%, in essence, it is not able to learn the task at all. Additionally, annealing (Algorithm 1) is clearly necessary: when started with the annealing algorithm's final and penultimate δ values, the model converged to E2EE values of 68% and 71% respectively.

J.3 SINGLE THRESHOLD

In this section, we consider a variation to the quantile function. We call this the *single threshold (ST)* sparsity approach, as opposed to the *multiple thresholds (MT)* technique described in Section 4.1, where we take the quantile across the entire input (batch axis, dimensional axes, channel axis). In this case, the channels can have differing resulting densities that average together to the target δ. More precisely, we use the quantile function $q_{\text{ST}} : \mathbb{R}^{B \times d_1 \times \ldots \times d_k \times n} \times \mathbb{R} \to \mathbb{R}$, implemented such that

$$p \approx \frac{1}{BSC} \sum_{b,i,c} \mathbf{1}(z[b,i,c] \leq q_{\text{ST}}(z,p)).$$

As seen in Figure 10, ST performs substantially worse in terms of CE and E2EE, while performing better with respect to FPE. Without the constraint that the motifs have equivalent density across each channel, some motifs are being used to represent multiple digits, which substantially increases confusion error, but also reduces false positives. In general, the MT model is superior as it has reasonable FPE and substantially lower CE/E2EE.

K COMPARISON TO DIRECTLY LEARNING THE MOTIFS

	SPARLING [mean]	DIRECT [mean]	Ratio [of means]
DigitCircle	1.24	0.01	0.01
LaTeX-OCR	6.55	0.12	0.02
LaTeX-OCR [without +()]	2.96	0.10	0.03
AudioMNISTSequence/train	5.41	0.61	0.11
AudioMNISTSequence/test	8.01	4.28	0.53

Table 2: Error [%] and ratios between errors. All are computed as a mean across 9 seeds

The purpose of SPARLING is to be able to learn intermediate state without having to have access to any training data on the intermediate state. In this section, we analyze how well it does at this goal, by comparing it to DIRECT, a setting where we train and evaluate on the intermediate state directly. Specifically, we construct datasets for each task of single motifs and train and test models on these datasets, then also test SPARLING on these datasets.

In the case of DIGITCIRCLE and LATEX-OCR, DIRECT is a trivial task as there is no distributional shift in the motif samples used to train and evaluate the model – effectively, DIRECT is tested on the training set. Thus, DIRECT gets $\sim$0% error.

However, on the AUDIOMNISTSEQUENCE task, the DIRECT has non-negligible error, with 0.61% error on the training sample distribution but a much higher 4.28% error on the testing sample distribution. Meanwhile, SPARLING increases substantially less, from 5.41% to 8.01%. This is because the error in SPARLING comes from two sources: the underlying uncertainty in prediction it shares with the DIRECT technique, and epistemic uncertainty related to the problem of identifying motifs from end-to-end data. This latter error evidently does not scale linearly with the difficulty of the underlying task.

L SPLICING DOMAIN

We also consider the original splicing domain, hypothesizing that on a domain that does not satisfy our assumptions in Section 3.3, SPARLING will not perform well but can perform better than chance. To keep things simple, we use the Jaganathan et al. (2019) architecture as the $\hat{h}$ model and a simple convolutional stack identical to the adjustment model from Gupta et al. (2024) as the $\hat{g}$ model. To ensure our experiment is picking up on a real signal, we will exclude the local splice site motifs (LSSI sites) from the set of true motifs for the purposes of analysis, as these sites can be found trivially from the end-to-end data, instead, we only evaluate on the other protein binding sites.

SPARLING achieves reasonable end-to-end performance, but does not perform as well as the other three domains on motif prediction (see Figure 11). However, we find that it consistently outperforms a random chance baseline in the most important error metric, CE—indicating that it is correctly classifying motifs. The other error metrics are more mixed, while it outperforms the baseline in FPE, it underperforms it in FNE suggesting that the model is producing duplicate activations, which leads to insufficient coverage of the motifs. Overall, this is consistent with our hypothesis that while Motif Identifiability is only possible given certain assumptions, SPARLING is capable of picking up some signal even when these assumptions are not met.

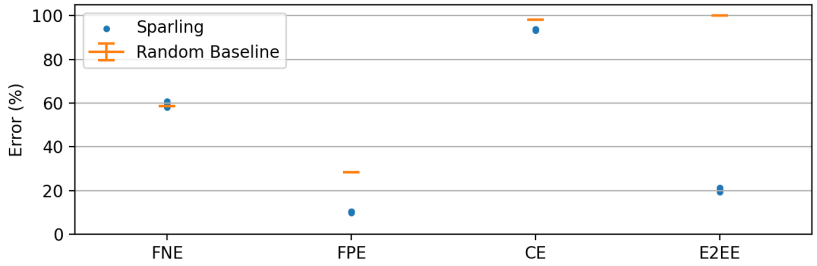

Figure 11: Results on the splicing domain. Results are presented per error metric for both 4 runs of SPARLING and 95% CI of a boostrap mean of 10 runs of a matched randomized baseline.

M COMPUTE USAGE

All our experiments were performed on NVIDIA GeForce GTX 1080 Ti (12GiB VRAM) or Quadro RTX 5000 (16GiB VRAM) GPUs. On average, DIGITCIRCLE experiments took 4 hours each to train, LATEX-OCR experiments took 14 days each to train, and AUDIOMNISTSEQUENCE experiments took 5 days to train. In total, we used about 350 GPU days of compute for the experiments reported in the body of the paper, 250 GPU days for the experiments referenced in footnotes/the appendix, and 200 GPU days of compute for exploratory experiments that were not referenced.