Atlas: Universal Function Approximator For Memory Retention—Supplementary Appendices

Anonymous Author(s)
Affiliation
Address
email

The supplementary appendices contain additional experiments (Section A) and proofs (Section B) to substantiate the claims in the main body of the paper.

A Additional experiments

The results for different experiments are given. Highlights from experiment A were discussed in the main paper, but the training and validation loss curves are only presented in the appendix.

Experimentation was performed with Python and TensorFlow. Experiment A was performed on a personal laptop with a 7th generation i7 Intel processor and took a few hours to finish thirty trials. The loss function chosen for training and evaluation is the mean absolute error (MAE). The training data set and test set in all experiments had 10000 data points, sampled uniformly at random. Gaussian noise with standard deviation $= 0.1$ was added to all training and test data target values. The test set was also used as a validation set to quantify the test error during training. All models were trained with a learning rate of 0.01 with the Adam optimizer. All models and experiments used batch sizes of 100 during training.

To test memory retention, two tasks, presented to an Atlas model one after the other, were constructed. The details of each task are given below.

Task 1 The training and test sets were sampled uniformly from the Task 1 target function over the domain $[0., 1.]^2$, with Gaussian noise added to the target values. The initial Atlas model was instantiated as a two-variable function that maps to a one-dimensional output, with $r = 0$ and $M = 0$ such that it is a minimally expressive model. The model was evaluated and trained for 30 epochs. After training, the Atlas model was expanded using the built-in methods, such that r is increased by one, and M is increased by two: $r' = r + 1$ and $M' = M + 2$. This training-expansion process was repeated four times. The output of the model is presented at the end of each expansion iteration. The target functions for Task 1 in each experiment is labeled $Y(\vec{\mathbf{x}})$.

Task 2 The test sets were sampled uniformly from the Task 2 target function over the domain $[0., 1.]^2$, with Gaussian noise added to the target values. The training sets were sampled uniformly over the domain $[0.45, 0.55]^2$, and target values of zero with added Gaussian noise. All models were trained for 6 epochs. The Task 2 target function in each experiment is given by:

$$Y'(\vec{\mathbf{x}}) = \begin{cases} 0 & 0.45 < x_i < 0.55 \; \forall i = 1, 2, ... \\ Y(\vec{\mathbf{x}}) & \text{otherwise.} \end{cases}$$

Task 2 effectively tests if a model changes only where new data was presented, with off-target effects leading to larger test MAE.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

A.1 Experiment A

A.1.1 Task 1

The Task 1 target function of Experiment A is given as follows, where the radius is measured from the centre of the domain $[0.5, 0.5]$:

$$r = \sqrt{(x_1 - \frac{1}{2})^2 + (x_2 - \frac{1}{2})^2}$$
$$\theta = \tan^{-1}\left((x - \frac{1}{2})^2, (y - \frac{1}{2})^2\right)$$
$$Y_A = Y_A(x_1, x_2) = \sin(30r + \theta) + 2$$

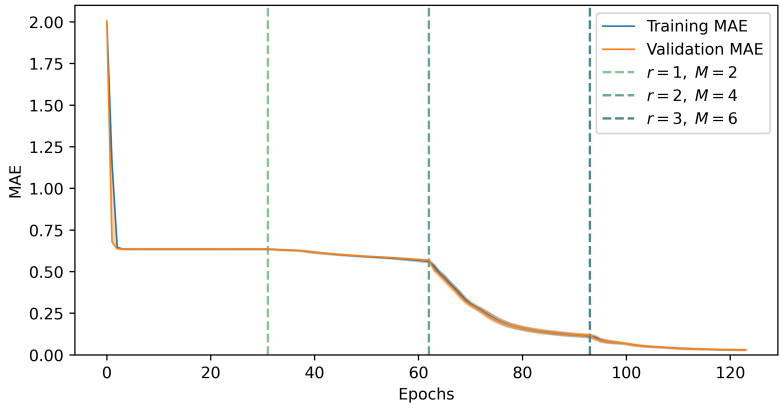

Figure 1: Training and validation MAE during the course of training on Task 1, Experiment A.

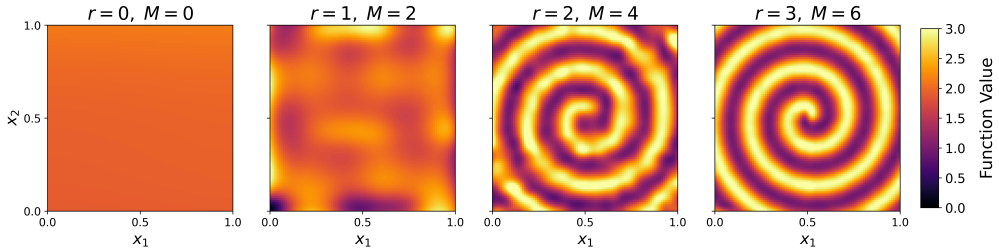

Figure 2: Outputs of the model during successive training and expansion iterations, Experiment A.

A.1.2 Task 2

The under-sampled target function Y'_A used for validation is given by:

$$Y'_A(x_1, x_2) = \begin{cases} 0 & 0.45 < x_i < 0.55 \ \forall i = 1, 2, ... \\ Y_A(x_1, x_2) & \text{otherwise.} \end{cases}$$

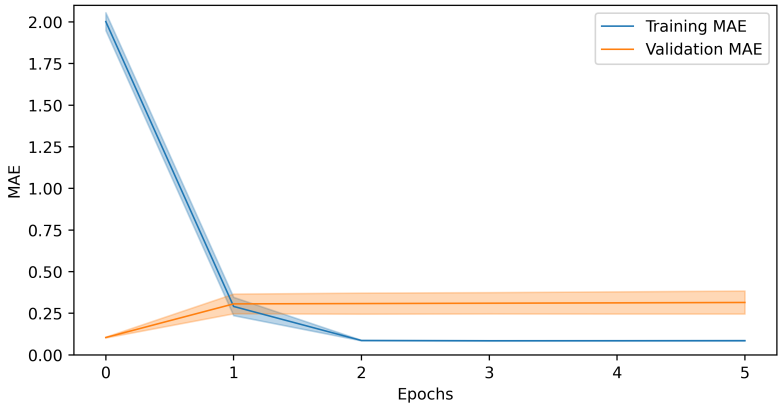

Figure 3: Training and validation MAE during the course of training on Task 2, Experiment A.

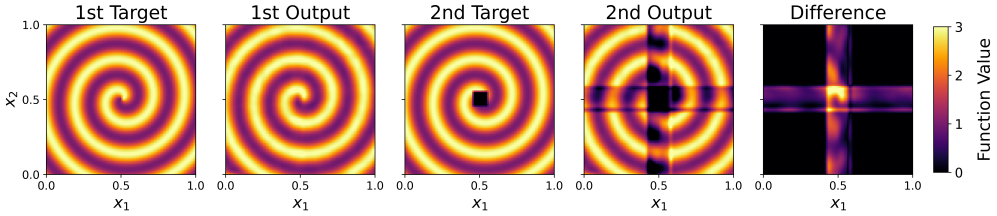

Figure 4: Visual inspection of target functions and model outputs over Task 1 and Task 2, Experiment A.

A.2 Experiment B

A.2.1 Task 1

The Task 1 target function for experiment B is given by:

$$Y_B(x_1, x_2) = \cos^2(20x_1 - 10) + \cos^2(10x_2 - 5) + \exp\left(-(20x_1 - 10)^2 - (20x_2 - 10)^2\right)$$

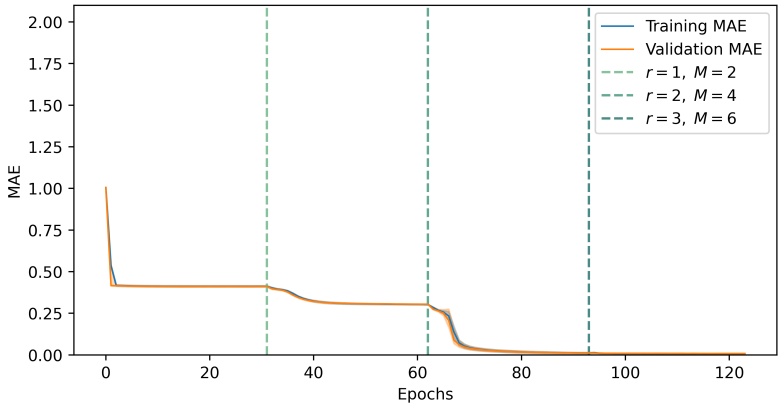

Figure 5: Training and validation MAE during the course of training on Task 1, Experiment B.

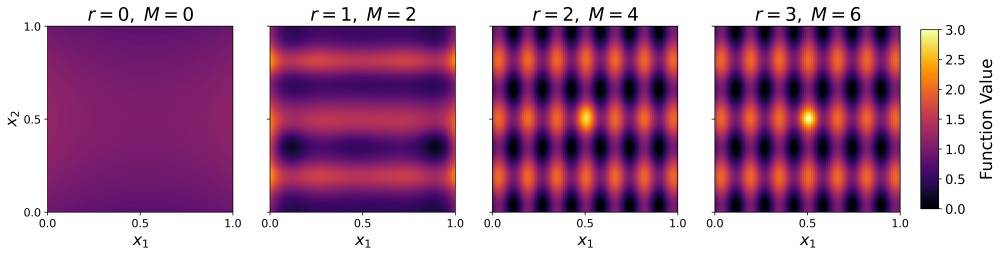

Figure 6: Outputs of the model during successive training and expansion iterations, Experiment B.

A.2.2 Task 2

The under-sampled target function Y'_B used for validation is given by:

$$Y'_B(x_1, x_2) = \begin{cases} 0 & 0.45 < x_i < 0.55 \; \forall i = 1, 2, ... \\ Y_B(x_1, x_2) & \text{otherwise.} \end{cases}$$

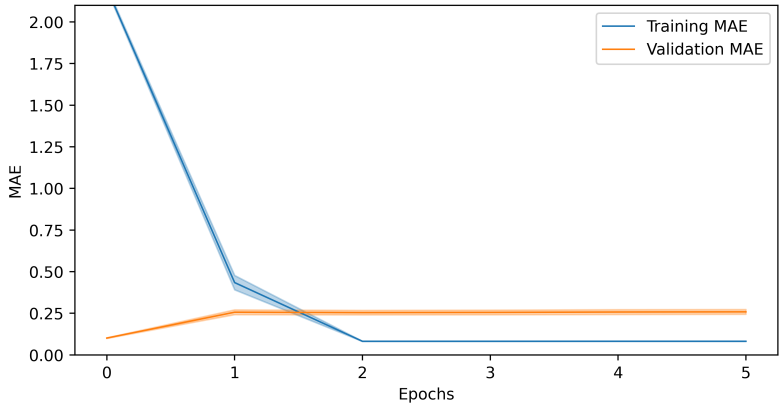

Figure 7: Training and validation MAE during the course of training on Task 2, Experiment B

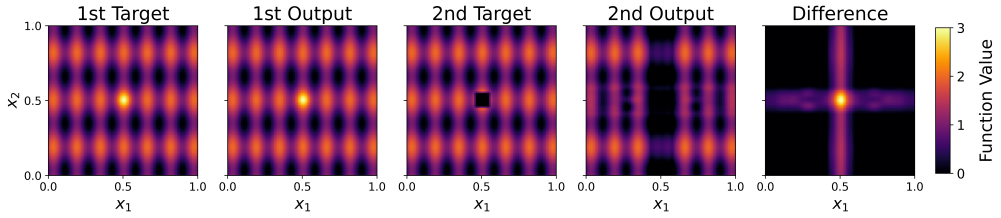

Figure 8: Visual inspection of target functions and model outputs over Task 1 and Task 2, Experiment B.

A.3 Experiment C

A.3.1 Task 1

The Task 1 target function for Experiment C is given by:

$$Y_C(x_1, x_2) = 2 + \cos(20x_1 - 10)\cos(20x_2 - 10)$$

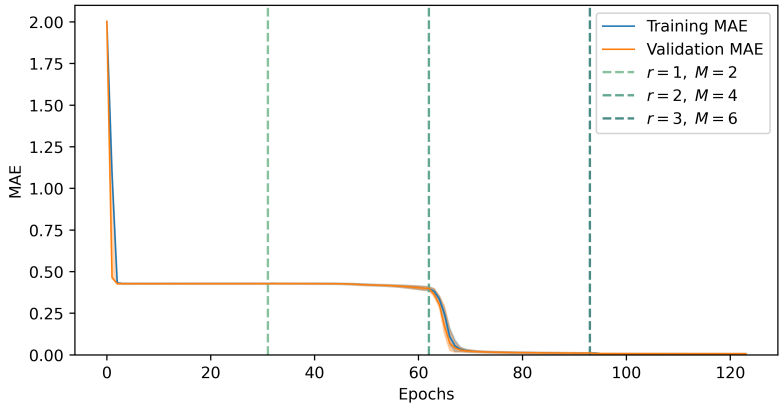

Figure 9: Training and validation MAE during the course of training on Task 1, Experiment C.

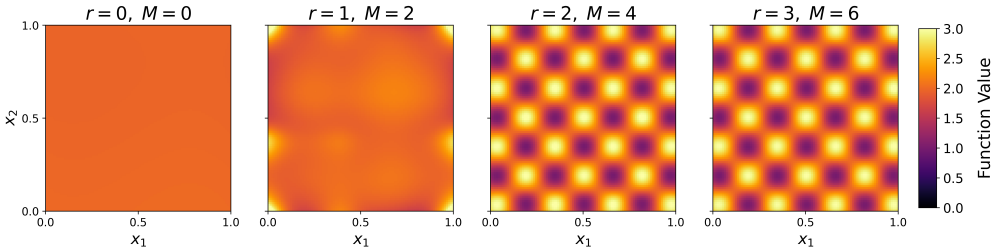

Figure 10: Outputs of the model during successive training and expansion iterations, Experiment C.

A.3.2 Task 2

The under-sampled target function Y'_C used for validation is given by:

$$Y'_C(x_1, x_2) = \begin{cases} 0 & 0.45 < x_i < 0.55 \ \forall i = 1, 2, ... \\ Y_C(x_1, x_2) & \text{otherwise.} \end{cases}$$

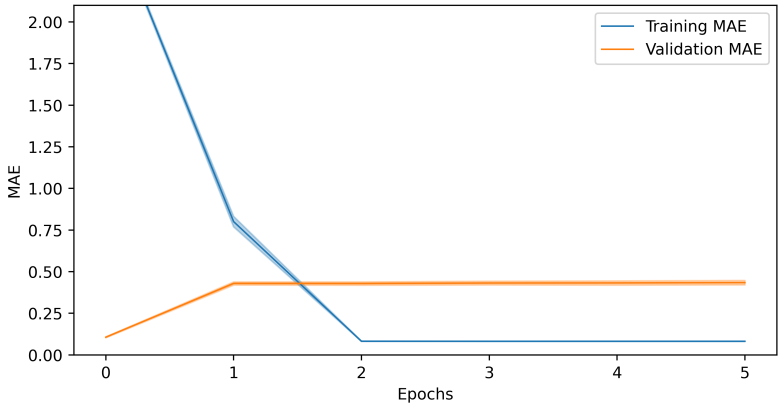

Figure 11: Training and validation MAE during the course of training on Task 2, Experiment C.

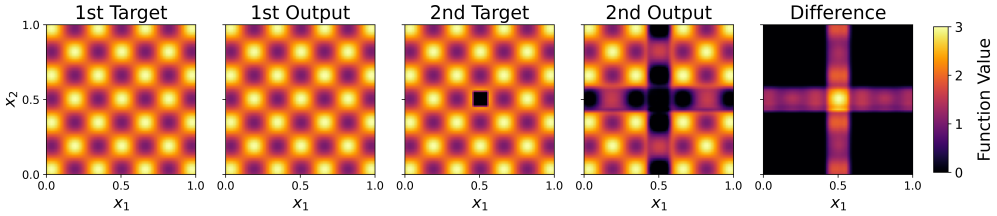

Figure 12: Visual inspection of target functions and model outputs over Task 1 and Task 2, Experiment C.

A.4 Experiment D

A.4.1 Task 1

The task 1 target function for experiment D is given by:

$$Y_D(x_1, x_2) = 2 + \sigma(\sin(2\pi x_1)\sin(2\pi x_2))$$

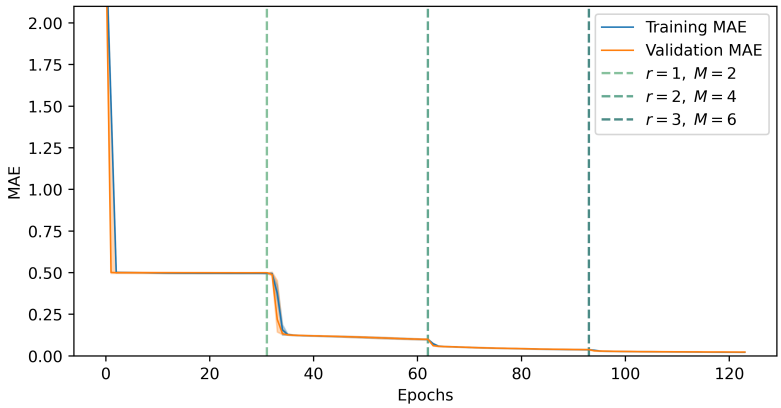

Figure 13: Training and validation MAE during the course of training on Task 1, Experiment D.

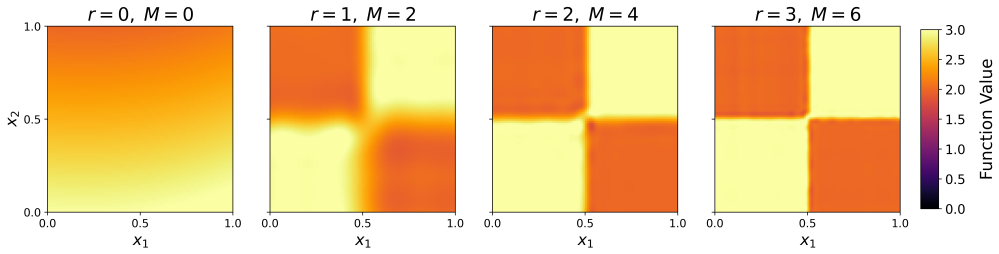

Figure 14: Outputs of the model during successive training and expansion iterations, Experiment D.

A.4.2 Task 2

The under-sampled target function Y'_D used for validation is given by:

$$Y'_D(x_1, x_2) = \begin{cases} 0 & 0.45 < x_i < 0.55 \ \forall i = 1, 2, ... \\ Y_D(x_1, x_2) & \text{otherwise.} \end{cases}$$

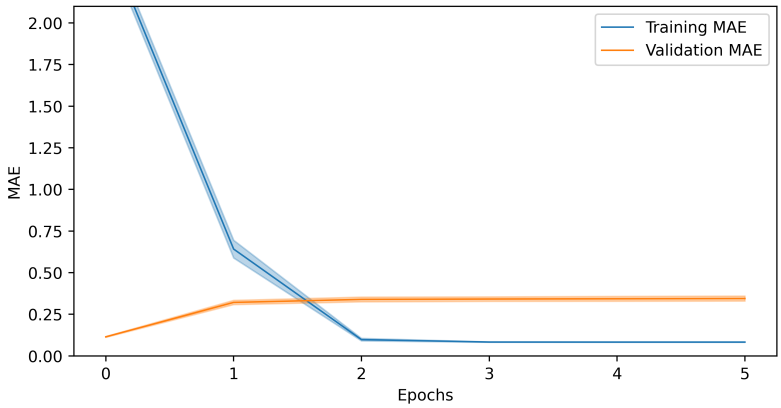

Figure 15: Training and validation MAE during the course of training on Task 2, Experiment D.

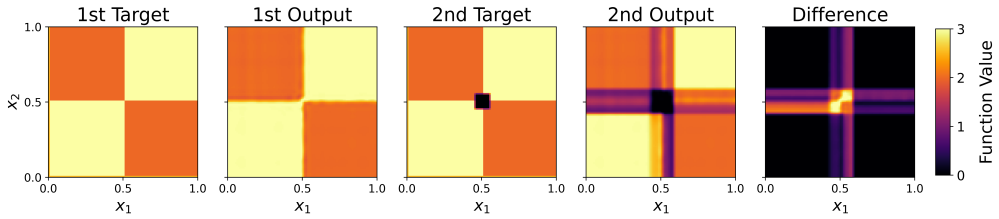

Figure 16: Visual inspection of target functions and model outputs over Task 1 and Task 2, Experiment D.

B Analysis and mathematical proofs

List of all definitions, theorems, corollaries, properties and their proofs are presented for completeness. The numbering of all statements match the main body of the paper. Some of the important results are also presented in the main body of the paper.

B.1 Stone-Weierstrass Theorem

Any continuous multi-variable function on a compact space can be uniformly approximated with multi-variable polynomials by the Stone-Weierstrass Theorem. Let $\mathcal{I}$ denote an index set of tuples of

natural numbers including zero such that $i_j \in \mathbb{N}^0$ for all $j \in \mathbb{N}$ with $i = (i_1, .., i_n) \in \mathcal{I}$ and $a_i \in \mathbb{R}$. Multi-variable polynomials can be represented as:

$$y(\vec{\mathbf{x}}) = y(x_1, .., x_n) = \sum_{i \in \mathcal{I}} a_i x_1^{i_1} x_2^{i_2} ... x_n^{i_n} = \sum_{i \in \mathcal{I}} a_i \Pi_{j=1}^n x_j^{i_j}$$

Each monomial term $a_i \Pi_{j=1}^n x_j^{i_j}$ is a product of single-variable functions in each variable.

B.2 Exponential Representation Theorem

Lemma 1. *For any $a_i \in \mathbb{R}$, there exists $\gamma_i > 0$ and $\beta_i > 0$, such that: $a_i = \gamma_i - \beta_i$*

Proof. Let $a_i \in \mathbb{R}$. Three cases are considered.

If $a_i = 0$, then choose $\gamma_i = 1 > 0$ and $\beta_i = 1 > 0$, such that: $\gamma_i - \beta_i = 1 - 1 = 0 = a_i$

If $a_i > 0$, then choose $\gamma_i = a_i + 1 > 0$ and $\beta_i = 1 > 0$, such that: $\gamma_i - \beta_i = a_i + 1 - 1 = a_i$

If $a_i < 0$, then choose $\gamma_i = 1 > 0$ and $\beta_i = 1 + |a_i| > 0$, such that:

$$\gamma_i - \beta_i = 1 - (1 + |a_i|) = 1 - 1 - |a_i| = a_i$$

□

Theorem 1 (Exponential representation theorem)**.** *Any multi-variable polynomial function $p(\vec{\mathbf{x}})$ of n variables over the positive orthant, can be exactly represented by continuous single-variable functions $g_{i,j}(x_j)$ and $h_{i,j}(x_j)$ in the form:*

$$p(\vec{\mathbf{x}}) = \sum_{i \in \mathcal{I}} \exp\big(\Sigma_{j=1}^n g_{i,j}(x_j)\big) - \exp\big(\Sigma_{j=1}^n h_{i,j}(x_j)\big)$$

Proof. Consider any monomial term $a_i \Pi_{j=1}^n x_j^{i_j}$ with $a_i \in \mathbb{R}$, then by Lemma 1 there exist strictly positive numbers $\gamma_i > 0$ and $\beta_i > 0$, such that:

$$\begin{aligned} a_i \Pi_{j=1}^n x_j^{i_j} &= \gamma_i \Pi_{j=1}^n x_j^{i_j} - \beta_i \Pi_{j=1}^n x_j^{i_j} \\ &= \exp\Big(\log\Big(\gamma_i \Pi_{j=1}^n x_j^{i_j}\Big)\Big) - \exp\Big(\log\Big(\beta_i \Pi_{j=1}^n x_j^{i_j}\Big)\Big) \\ &= \exp\Big(\log(\gamma_i) + \Sigma_{j=1}^n \log\Big(x_j^{i_j}\Big)\Big) - \exp\Big(\log(\beta_i) + \Sigma_{j=1}^n \log\Big(x_j^{i_j}\Big)\Big) \end{aligned}$$

The argument of each exponential function is a sum of single-variable functions and constants. Without loss of generality, a set of single-variable functions can be defined such that:

$$a_i \Pi_{j=1}^n x_j^{i_j} = \exp\big(\Sigma_{j=1}^n g_{i,j}(x_j)\big) - \exp\big(\Sigma_{j=1}^n h_{i,j}(x_j)\big)$$

Since this holds for any $a_i \Pi_{j=1}^n x_j^{i_j}$ and all $i \in \mathcal{I}$, it follows that:

$$p(\vec{\mathbf{x}}) = \sum_{i \in \mathcal{I}} \exp\big(\Sigma_{j=1}^n g_{i,j}(x_j)\big) - \exp\big(\Sigma_{j=1}^n h_{i,j}(x_j)\big)$$

□

There is a duality between representation and approximation. If any multi-variable polynomial can be exactly represented, then any continuous multi-variable function can be approximated to arbitrary accuracy.

B.2.1 Exponential approximation corollary

Corollary 1 (Exponential approximation)**.** *For any $\varepsilon > 0$, and continuous multi-variable function $y(\vec{\mathbf{x}})$ of n variables over a compact domain in the positive orthant, there exist continuous single-variable functions $g_{i,j}(x_j)$ and $h_{i,j}(x_j)$ such that:*

$$\left| y(\vec{\mathbf{x}}) - \left(\sum_{i \in \mathcal{I}} \exp\left(\Sigma_{j=1}^{n} g_{i,j}(x_j)\right) - \exp\left(\Sigma_{j=1}^{n} h_{i,j}(x_j)\right) \right) \right| < \varepsilon$$

Proof. Fix $\varepsilon > 0$, and let $y(\vec{\mathbf{x}})$ be a continuous multi-variable function of n variables over a compact domain in the positive orthant.

By the Stone–Weierstrass theorem there exists a multi-variable polynomial $p(\vec{\mathbf{x}})$ over the domain of $y(\vec{\mathbf{x}})$ such that:

$$|y(\vec{\mathbf{x}}) - p(\vec{\mathbf{x}})| < \varepsilon$$

By Theorem 1, for any polynomial $p(\vec{\mathbf{x}})$ over the positive orthant, there exist continuous single-variable functions $g_{i,j}(x_j)$ and $h_{i,j}(x_j)$ such that:

$$p(\vec{\mathbf{x}}) = \sum_{i \in \mathcal{I}} \exp\left(\Sigma_{j=1}^{n} g_{i,j}(x_j)\right) - \exp\left(\Sigma_{j=1}^{n} h_{i,j}(x_j)\right)$$

It follows that:

$$\left| y(\vec{\mathbf{x}}) - \left(\sum_{i \in \mathcal{I}} \exp\left(\Sigma_{j=1}^{n} g_{i,j}(x_j)\right) - \exp\left(\Sigma_{j=1}^{n} h_{i,j}(x_j)\right) \right) \right| < \varepsilon$$

□

B.3 Single-variable function approximators

Each basis function S_i for a uniform cubic B-spline can be obtained by scaling and translating the input of the same activation function. The activation function is denoted $S(x)$ and is given by:

$$S(x) = \begin{cases} \frac{1}{6}x^3 & 0 \le x < 1 \\ \frac{1}{6}\left[-3(x-1)^3 + 3(x-1)^2 + 3(x-1) + 1\right] & 1 \le x < 2 \\ \frac{1}{6}\left[3(x-2)^3 - 6(x-2)^2 + 4\right] & 2 \le x < 3 \\ \frac{1}{6}(4-x)^3 & 3 \le x < 4 \\ 0 & otherwise \end{cases}$$

B.3.1 Definition of ρ-density B-spline functions

Definition 1 (ρ-density B-spline function)**.** A ρ-density B-spline function is a uniform cubic B-spline function with $2^{\rho+2}$ basis functions:

$$f(x) = \sum_{i=1}^{2^{\rho+2}} \theta_i S_i(x) = \sum_{i=1}^{2^{\rho+2}} \theta_i S(w_i x + b_i) = \sum_{i=1}^{2^{\rho+2}} \theta_i S((2^{\rho+2} - 3)x + 4 - i)$$

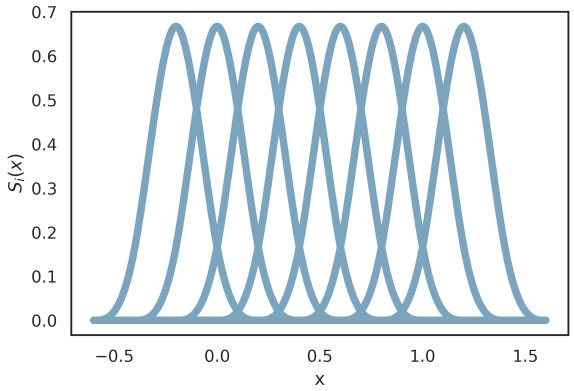

Figure 17: Set of eight uniform cubic B-spline basis functions, where $S_i(x) = S(w_i x + b_i)$.

B.3.2 Definition of mixed-density B-spline function

Definition 2 (mixed-density B-spline function)**.** A mixed-density B-spline function is a single-variable function approximator that is obtained by summing together different ρ-density B-spline functions. Only the maximum ρ-density B-spline function has trainable parameters, the others are constant. Mixed-density B-spline functions are of the form:

$$f(x) = \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{\rho,i} S_{\rho,i}(x)$$

The maximum density parameters $\theta_{r,i}$ are trainable, but the lower density parameters $\theta_{\rho,i}$ (with $\rho < r$) are in general non-zero constants. The function approximator can be expanded without losing previously learned values. Analytically, we can choose all the new parameters $\theta_{r+1,i} = 0,\ \forall i \in \mathbf{N}$ such that:

$$f(x) = \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{\rho,i} S_{\rho,i}(x) = \sum_{\rho=0}^{r+1} \sum_{i=1}^{2^{\rho+2}} \theta_{\rho,i} S_{\rho,i}(x)$$

B.4 Atlas architecture

B.4.1 Atlas representation theorem

Theorem 2 (Atlas representation theorem)**.** *Any multi-variable polynomial $p(\vec{\mathbf{x}})$ of n variables over the positive orthant, can be exactly represented by continuous single-variable functions $f_j(x_j)$, $g_{i,j}(x_j)$, and $h_{i,j}(x_j)$ in the form:*

$$p(\vec{\mathbf{x}}) = \sum_{j=1}^{n} f_j(x_j) + \sum_{k=1}^{\infty} \frac{1}{k^2} \exp\left(\Sigma_{j=1}^{n} g_{k,j}(x_j)\right) - \frac{1}{k^2} \exp\left(\Sigma_{j=1}^{n} h_{k,j}(x_j)\right)$$

Proof. Let $p(\vec{\mathbf{x}})$ be a multi-variable polynomial over the positive orthant:

$$p(\vec{\mathbf{x}}) = p(x_1, .., x_n) = \sum_{i \in \mathcal{I}} a_i x_1^{i_1} x_2^{i_2} ... x_n^{i_n} = \sum_{i \in \mathcal{I}} a_i \Pi_{j=1}^{n} x_j^{i_j}$$

Consider the set of terms that depend on at most one input variable, or single-variable terms in the expression for the polynomial $p(\vec{\mathbf{x}})$:

$$\mathcal{P}_1 := \{a_i \Pi_{j=1}^{n} x_j^{i_j} \mid i \in \mathcal{I},\ i_j \neq 0 \implies i_k = 0,\ \forall k \neq j\}$$

It is worth noting that $\mathcal{P}_1$ contains the constant function.

Let $\mathcal{Q}$ denote the index set of all single-variable monomial terms:

$$\mathcal{Q} := \{\, i \mid i \in \mathcal{I},\ i_j \neq 0 \implies i_k = 0,\ \forall k \neq j \}$$

The polynomial $p(\vec{\mathbf{x}})$ can be rewritten in terms of single-variable functions and a residual polynomial function p_{res} as:

$$\begin{aligned} p(\vec{\mathbf{x}}) &= \sum_{i \in \mathcal{Q}} a_i \Pi_{j=1}^{n} x_j^{i_j} + \sum_{i \in \mathcal{I} \setminus \mathcal{Q}} a_i \Pi_{j=1}^{n} x_j^{i_j} \\ &= \sum_{j=1}^{n} f_j(x_j) + p_{res}(\vec{\mathbf{x}}) \end{aligned}$$

The single-variable terms can be consumed by a sum of n arbitrary single-variable functions $f_j(x_j)$.

By Theorem 1, for any polynomial $p_{res}(\vec{\mathbf{x}})$ over the positive orthant, there exist continuous single-variable functions $g_{i,j}(x_j)$ and $h_{i,j}(x_j)$ such that:

$$p_{res}(\vec{\mathbf{x}}) = \sum_{i \in \mathcal{I} \setminus \mathcal{Q}} \exp\big(\Sigma_{j=1}^{n} g_{i,j}(x_j)\big) - \exp\big(\Sigma_{j=1}^{n} h_{i,j}(x_j)\big)$$

Since the index set is countable, one can use another indexing scheme:

$$p_{res}(\vec{\mathbf{x}}) = \sum_{k=1}^{\infty} \exp\big(\Sigma_{j=1}^{n} g_{k,j}(x_j)\big) - \exp\big(\Sigma_{j=1}^{n} h_{k,j}(x_j)\big)$$

Scale factors can be introduced without changing the representation:

$$\begin{aligned} p_{res}(\vec{\mathbf{x}}) &= \sum_{k=1}^{\infty} \exp\big(\log k^2 - \log k^2 + \Sigma_{j=1}^{n} g_{k,j}(x_j)\big) - \exp\big(\log k^2 - \log k^2 + \Sigma_{j=1}^{n} h_{k,j}(x_j)\big) \\ &= \sum_{k=1}^{\infty} \frac{1}{k^2} \exp\big(\log k^2 + \Sigma_{j=1}^{n} g_{k,j}(x_j)\big) - \frac{1}{k^2} \exp\big(\log k^2 + \Sigma_{j=1}^{n} h_{k,j}(x_j)\big) \end{aligned}$$

Since the single-variable functions $g_{i,j}(x_j)$ and $h_{i,j}(x_j)$ are arbitrary, one can absorb the constants and redefine $g_{i,j}(x_j)$ and $h_{i,j}(x_j)$ to obtain:

$$p_{res}(\vec{\mathbf{x}}) = \sum_{k=1}^{\infty} \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} g_{k,j}(x_j)\big) - \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} h_{k,j}(x_j)\big)$$

Substituting the expressions one obtains:

$$p(\vec{\mathbf{x}}) = \sum_{j=1}^{n} f_j(x_j) + \sum_{k=1}^{\infty} \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} g_{k,j}(x_j)\big) - \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} h_{k,j}(x_j)\big)$$

□

There is a duality between representation and approximation. If any multi-variable polynomial can be exactly represented, then any continuous multi-variable function can be approximated to arbitrary accuracy.

B.4.2 Atlas definition

Definition 3 (Atlas)**.** Atlas is a function approximator of n variables, with mixed-density B-spline functions $f_j(x_j)$, $g_{i,j}(x_j)$, and $h_{i,j}(x_j)$ in the form:

$$A(\vec{\mathbf{x}}) := \sum_{j=1}^{n} f_j(x_j) + \sum_{k=1}^{M} \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} g_{k,j}(x_j)\big) - \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} h_{k,j}(x_j)\big)$$

Atlas is equivalently given by the compact notation:

$$\begin{aligned} A(\vec{\mathbf{x}}) :=& \sum_{j=1}^{n} f_j(x_j) + \sum_{k=1}^{M} \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} g_{k,j}(x_j)\big) - \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} h_{k,j}(x_j)\big) \\ =& F(\vec{\mathbf{x}}) + \sum_{k=1}^{M} \frac{1}{k^2} \exp(G_k(\vec{\mathbf{x}})) - \frac{1}{k^2} \exp(H_k(\vec{\mathbf{x}})) \\ =& F(\vec{\mathbf{x}}) + G(\vec{\mathbf{x}}) - H(\vec{\mathbf{x}}) \end{aligned}$$

B.4.3 Atlas polynomial approximation

Theorem 3 (Atlas polynomial approximation)**.** *For any multi-variable polynomial $p(\vec{\mathbf{x}})$ over the positive orthant with bounded and compact domain $D(p)$ and $\varepsilon > 0$, there exists an Atlas model $A(\vec{\mathbf{x}})$ such that:*

$$|p(\vec{\mathbf{x}}) - A(\vec{\mathbf{x}})| < \varepsilon$$

Proof. Let $p(\vec{\mathbf{x}})$ be a multi-variable polynomial $p(\vec{\mathbf{x}})$ of n variables over the positive orthant, and fix $\varepsilon > 0$, and choose:

$$\varepsilon = \varepsilon_1 + \varepsilon_2$$

By Theorem 2, there exist continuous single-variable functions $f_j(x_j)$, $g_{i,j}(x_j)$, and $h_{i,j}(x_j)$ such that:

$$p(\vec{\mathbf{x}}) = \sum_{j=1}^{n} f_j(x_j) + \sum_{k=1}^{\infty} \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} g_{k,j}(x_j)\big) - \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} h_{k,j}(x_j)\big)$$

If $p(\vec{\mathbf{x}})$ has finitely many terms, then let M denote the number of residual terms:

$$\begin{aligned} p(\vec{\mathbf{x}}) =& \sum_{j=1}^{n} f_j(x_j) + \sum_{k=1}^{M} \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} g_{k,j}(x_j)\big) - \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} h_{k,j}(x_j)\big) \\ =& F(\vec{\mathbf{x}}) + \sum_{k=1}^{M} \frac{1}{k^2} \exp(G_k(\vec{\mathbf{x}})) - \frac{1}{k^2} \exp(H_k(\vec{\mathbf{x}})) \\ =& F(\vec{\mathbf{x}}) + G(\vec{\mathbf{x}}) - H(\vec{\mathbf{x}}) \end{aligned}$$

Choose mixed-density B-spline functions $f_j^*(x_j)$, $g_{i,j}^*(x_j)$, and $h_{i,j}^*(x_j)$ such that the Atlas model $A(\vec{\mathbf{x}})$ is given by:

$$
\begin{aligned}
A^*(\vec{\mathbf{x}}) =& \sum_{j=1}^{n} f_j^*(x_j) + \sum_{k=1}^{M} \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} g_{k,j}^*(x_j)\big) - \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} h_{k,j}^*(x_j)\big) \\
=& F^*(\vec{\mathbf{x}}) + \sum_{k=1}^{M} \frac{1}{k^2} \exp(G_k^*(\vec{\mathbf{x}})) - \frac{1}{k^2} \exp(H_k^*(\vec{\mathbf{x}})) \\
=& F^*(\vec{\mathbf{x}}) + G^*(\vec{\mathbf{x}}) - H^*(\vec{\mathbf{x}})
\end{aligned}
$$

Then it follows that,

$$
\begin{aligned}
|p(\vec{\mathbf{x}}) - A(\vec{\mathbf{x}})| =& |F(\vec{\mathbf{x}}) + G(\vec{\mathbf{x}}) - H(\vec{\mathbf{x}}) - (F^*(\vec{\mathbf{x}}) + G^*(\vec{\mathbf{x}}) - H^*(\vec{\mathbf{x}}))| \\
=& |F(\vec{\mathbf{x}}) - F^*(\vec{\mathbf{x}}) + G(\vec{\mathbf{x}}) - G^*(\vec{\mathbf{x}}) - (H(\vec{\mathbf{x}}) - H^*(\vec{\mathbf{x}}))| \\
\leq& |F(\vec{\mathbf{x}}) - F^*(\vec{\mathbf{x}})| + |G(\vec{\mathbf{x}}) - G^*(\vec{\mathbf{x}})| + |H(\vec{\mathbf{x}}) - H^*(\vec{\mathbf{x}})|
\end{aligned}
$$

The first set of functions is easily shown to have bounded error. Choose mixed-density B-spline functions $f_j^*(x_j)$ such that:

$$
|f_j(x_j) - f_j^*(x_j)| < \frac{\varepsilon_1}{n}
$$

Then it follows,

$$
\begin{aligned}
|F(\vec{\mathbf{x}}) - F^*(\vec{\mathbf{x}})| =& \left| \sum_{j=1}^{n} f_j(x_j) - \sum_{j=1}^{n} f_j^*(x_j) \right| \\
\leq& \sum_{j=1}^{n} \left| f_j(x_j) - f_j^*(x_j) \right| \\
<& \varepsilon_1
\end{aligned}
$$

The interior functions for the exponential functions are more complicated.

Remark. The uniform continuity of exponentials on bouned domains makes it possible to bound the approximation error in each exponential term. The exponential function is uniformly continuous on a compact and bounded subset of the real numbers $[a, b]$. Thus, for any $\varepsilon_{\exp} > 0$, there exists a $\delta_{\exp} > 0$, such that for every $x, y \in [a, b]$:

$$
|x - y| < \delta_{\exp} \implies |\exp(x) - \exp(y)| < \varepsilon_{\exp}
$$

For all exponential functions on bounded and compact domains choose:

$$
\varepsilon_{\exp} = \frac{3\varepsilon_2}{\pi^2}
$$

Choose the smallest $\delta_{\exp}$ for all M exponential functions, so that the implication holds. Choose $\delta_{g,k,j}$, such that:

$$
\sum_{j=1}^{n} \delta_{g,k,j} < \delta_{\exp}
$$

Choose mixed-density B-spline functions $g_{i,j}^*(x_j)$, and $h_{i,j}^*(x_j)$ such that:

$$|g_{k,j}(x_j) - g^*_{k,j}(x_j)| < \delta_{g,k,j}$$
$$|h_{k,j}(x_j) - h^*_{k,j}(x_j)| < \delta_{g,k,j}$$

The interior functions $g^*_{k,j}(x_j)$ have bounded approximation error $\delta_{g,k,j}$ one obtains:

$$\begin{aligned}
&\left|\sum_{j=1}^{n} g_{k,j}(x_j) - \sum_{j=1}^{n} g^*_{k,j}(x_j)\right| \\
\leq &\sum_{j=1}^{n} \left|g_{k,j}(x_j) - g^*_{k,j}(x_j)\right| \\
< &\sum_{j=1}^{n} \delta_{g,k,j} < \delta_{\exp}
\end{aligned}$$

This implies that:

$$\left|\exp\left(\Sigma_{j=1}^{n} g_{k,j}(x_j)\right) - \exp\left(\Sigma_{j=1}^{n} g^*_{k,j}(x_j)\right)\right| < \varepsilon_{\exp}$$

Recombining this result with the exponential terms yields:

$$\begin{aligned}
|G(\vec{\mathbf{x}}) - G^*(\vec{\mathbf{x}})| = &\left|\sum_{k=1}^{M} \frac{1}{k^2} \exp\left(\Sigma_{j=1}^{n} g_{k,j}(x_j)\right) - \sum_{k=1}^{M} \frac{1}{k^2} \exp\left(\Sigma_{j=1}^{n} g^*_{k,j}(x_j)\right)\right| \\
|G(\vec{\mathbf{x}}) - G^*(\vec{\mathbf{x}})| \leq &\sum_{k=1}^{M} \frac{1}{k^2} \left|\exp\left(\Sigma_{j=1}^{n} g_{k,j}(x_j)\right) - \exp\left(\Sigma_{j=1}^{n} g^*_{k,j}(x_j)\right)\right| \\
|G(\vec{\mathbf{x}}) - G^*(\vec{\mathbf{x}})| < &\sum_{k=1}^{M} \frac{1}{k^2} \varepsilon_{\exp}
\end{aligned}$$

The scaling factors of k^{-2} were chosen for convergence, such that:

$$\begin{aligned}
|G(\vec{\mathbf{x}}) - G^*(\vec{\mathbf{x}})| < &\sum_{k=1}^{\infty} \frac{1}{k^2} \varepsilon_{\exp} \\
|G(\vec{\mathbf{x}}) - G^*(\vec{\mathbf{x}})| < &\varepsilon_{\exp} \sum_{k=1}^{\infty} \frac{1}{k^2} \\
|G(\vec{\mathbf{x}}) - G^*(\vec{\mathbf{x}})| < &\varepsilon_{\exp} \frac{\pi^2}{6} = \frac{3\varepsilon_2}{\pi^2} \frac{\pi^2}{6} = \frac{\varepsilon_2}{2}
\end{aligned}$$

It follows that:

$$|G(\vec{\mathbf{x}}) - G^*(\vec{\mathbf{x}})| < \frac{\varepsilon_2}{2}$$

The same argument holds for $|H(\vec{\mathbf{x}}) - H^*(\vec{\mathbf{x}})|$, and one obtains the result:

$$\begin{aligned}
|p(\vec{\mathbf{x}}) - A(\vec{\mathbf{x}})| \leq &|F(\vec{\mathbf{x}}) - F^*(\vec{\mathbf{x}})| + |G(\vec{\mathbf{x}}) - G^*(\vec{\mathbf{x}})| + |H(\vec{\mathbf{x}}) - H^*(\vec{\mathbf{x}})| \\
|p(\vec{\mathbf{x}}) - A(\vec{\mathbf{x}})| < &\varepsilon_1 + \frac{\varepsilon_2}{2} + \frac{\varepsilon_2}{2} \\
|p(\vec{\mathbf{x}}) - A(\vec{\mathbf{x}})| < &\varepsilon_1 + \varepsilon_2
\end{aligned}$$

Finally,

$$|p(\vec{\mathbf{x}}) - A(\vec{\mathbf{x}})| < \varepsilon$$

□

B.4.4 Universal function approximation theorem

Theorem 4 (Atlas universal function approximation)**.** *For any $\varepsilon > 0$, and continuous multi-variable function $y(\vec{\mathbf{x}})$ of n variables over a compact domain in the positive orthant, there exists an Atlas model $A(\vec{\mathbf{x}})$ such thtat:*

$$|y(\vec{\mathbf{x}}) - A(\vec{\mathbf{x}})| < \varepsilon$$

Proof. Let $\varepsilon > 0$, and $y(\vec{\mathbf{x}})$ be a continuous multi-variable function of n variables over a compact domain in the positive orthant. Choose $\varepsilon_1 + \varepsilon_2 = \varepsilon$.

By the Stone–Weierstrass theorem there exists a multi-variable polynomial $p(\vec{\mathbf{x}})$ over the domain of $y(\vec{\mathbf{x}})$ such that:

$$|y(\vec{\mathbf{x}}) - p(\vec{\mathbf{x}})| < \varepsilon_1$$

By Theorem 3 an Atlas model can approximate the polynomial $p(\vec{\mathbf{x}})$ to arbitrary precision:

$$|p(\vec{\mathbf{x}}) - A(\vec{\mathbf{x}})| < \varepsilon_2$$

It follows from the triangle inequality that:

$$\begin{aligned} |y(\vec{\mathbf{x}}) - A(\vec{\mathbf{x}})| &\leq |y(\vec{\mathbf{x}}) - p(\vec{\mathbf{x}})| + |p(\vec{\mathbf{x}}) - A(\vec{\mathbf{x}})| \\ |y(\vec{\mathbf{x}}) - A(\vec{\mathbf{x}})| &< \varepsilon_1 + \varepsilon_2 \end{aligned}$$

Finally,

$$|y(\vec{\mathbf{x}}) - A(\vec{\mathbf{x}})| < \varepsilon$$

□

B.5 Atlas properties

B.5.1 Atlas expansion

The number of exponential terms can be increased without changing the output of the model. We can choose to initialise $G_{M+1}(\vec{\mathbf{x}}) = 0$ and $H_{M+1}(\vec{\mathbf{x}}) = 0$, such that the model capacity can be increased without changing the output of the model:

$$\begin{aligned} A_{M+1}(\vec{\mathbf{x}}) &= \sum_{k=1}^{M+1} \frac{1}{k^2} \exp(G_k(\vec{\mathbf{x}})) - \frac{1}{k^2} \exp(H_k(\vec{\mathbf{x}})) \\ &= \frac{1}{(M+1)^2} \exp(G_{M+1}(\vec{\mathbf{x}})) - \frac{1}{(M+1)^2} \exp(H_{M+1}(\vec{\mathbf{x}})) + A(\vec{\mathbf{x}}) \\ &= \frac{1}{(M+1)^2} \exp(0) - \frac{1}{(M+1)^2} \exp(0) + A(\vec{\mathbf{x}}) \\ &= A(\vec{\mathbf{x}}) \end{aligned}$$

The density of basis functions in Atlas can also be incremented without changing the learned output of the model. The density of basis functions can also be increased without changing the output for any of mixed-density B-spline functions Ψ of the form:

$$\Psi(x) = \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{\rho,i} S_{\rho,i}(x)$$

Analytically, we can choose all the new parameters $\theta_{r+1,i} = 0, \ \forall i \in \mathbf{N}$ such that:

$$\Psi(x) = \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{\rho,i} S_{\rho,i}(x) = \sum_{\rho=0}^{r+1} \sum_{i=1}^{2^{\rho+2}} \theta_{\rho,i} S_{\rho,i}(x)$$

The last thing to note is that only the parameters for the largest specified density are trainable, in contrast to smaller density parameters that are fixed constants.

B.5.2 Atlas sparsity

Property 1 (Sparsity)**.** For any $\vec{\mathbf{x}} \in D(A) \subset R^n$ and bounded trainable parameters θ_i with index set Θ, the gradient vector of trainable parameters for Atlas is sparse:

$$\left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} A(\vec{\mathbf{x}}) \right\|_0 = \sum_{i \in \Theta} d_{Hamming} \left(\frac{\partial A}{\partial \theta_i}(\vec{\mathbf{x}}), 0 \right) \leq 4n(2M+1)$$

Proof. Let $A(\vec{\mathbf{x}})$ denote some Atlas model, with mixed-density B-spline functions $f_j(x_j)$, $g_{i,j}(x_j)$, and $h_{i,j}(x_j)$ in the form:

$$A(\vec{\mathbf{x}}) = \sum_{j=1}^{n} f_j(x_j) + \sum_{k=1}^{M} \frac{1}{k^2} \exp\left(\Sigma_{j=1}^{n} g_{k,j}(x_j)\right) - \frac{1}{k^2} \exp\left(\Sigma_{j=1}^{n} h_{k,j}(x_j)\right)$$

Each mixed-density B-spline function has its own parameters that are independent of every other mixed-density B-spline. The mixed-density B-splines function $\Psi(x)$ is by definition given by:

$$\Psi(x) = \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{\rho,i} S_{\rho,i}(x)$$

Thus for every mixed-density B-spline function in $A(\vec{\mathbf{x}})$:

$$f_j(x_j) = \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{f,(\rho,i,j)} S_{\rho,i}(x_j)$$

$$g_{k,j}(x_j)) = \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{g,(\rho,i,k,j)} S_{\rho,i}(x)$$

$$h_{k,j}(x_j)) = \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{h,(\rho,i,k,j)} S_{\rho,i}(x)$$

Only the maximum density basis function $\rho = r$ have trainable parameters. The maximum density r-density B-spline functions are uniform B-spline functions with trainable parameters. There are at most four basis functions that are non-zero for any given x_j, and as such the gradient vector with respect to trainable parameters will have at most four non-zero entries for each r-density B-spline

function, the same four parameters for each mixed-density B-spline functions $f_j(x_j)$, $g_{i,j}(x_j)$, and $h_{i,j}(x_j)$. One simply needs to count the number of mixed-density B-spline functions.

The number of mixed-density B-spline functions labeled $f_j(x_j)$ are in total n, with 4 active trainable parameters each.

The number of functions labeled $g_{k,j}(x_j)$ are in total nM. For each M, there are n mixed-density B-spline functions, with 4 active trainable parameters each.

The number of mixed-density B-spline functions labeled $h_{k,j}(x_j)$ are in total nM. For each M, there are n mixed-density B-spline functions, with 4 active trainable parameters each.

The total number of active trainable parameters is thus:

$$4n + 4nM + 4nM = 4n(2M + 1)$$

□

Remark. The total number of trainable parameters for each mixed-density B-spline function is 2^{r+2}. For a fixed number of variables n, the model has a total of $2^{r+2}n(2M + 1)$ trainable parameters. The gradient vector has a maximum of $4n(2M + 1)$ non-zero entries, which is independent of r. Recall that only the maximum density ($\rho = r$) cubic B-spline function has trainable parameters. The fraction of trainable basis functions that are active is at most 2^{-r}. Sparsity entails efficient implementation, and suggests possible memory retention and robustness to catastrophic forgetting.

It is worth noting that the total number of parameters (including constants) is:

$$\text{Total number of parameters} \propto \sum_{\rho=0}^{r} 2^{\rho+2}n(2M + 1) \approx 2^{r+1}n(2M + 1)$$

B.5.3 Atlas gradient flow attenuation

Property 2 (Gradient flow attenuation)**.** For any $\vec{\mathbf{x}} \in D(A) \subset R^n$ and bounded trainable parameters θ_i with index set Θ: if all the mixed-density B-spline functions are bounded, then the gradient vector of trainable parameters for Atlas is bounded:

$$\left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} A(\vec{\mathbf{x}}) \right\|_1 = \sum_{i \in \Theta} \left| \frac{\partial A}{\partial \theta_i}(\vec{\mathbf{x}}) \right| < U$$

Proof. Let $A(\vec{\mathbf{x}})$ denote some Atlas model, with mixed-density B-spline functions $f_j(x_j)$, $g_{i,j}(x_j)$, and $h_{i,j}(x_j)$ in the form:

$$\begin{aligned} A(\vec{\mathbf{x}}) =& \sum_{j=1}^{n} f_j(x_j) + \sum_{k=1}^{M} \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} g_{k,j}(x_j)\big) - \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} h_{k,j}(x_j)\big) \\ =& F(\vec{\mathbf{x}}) + \sum_{k=1}^{M} \frac{1}{k^2} \exp(G_k(\vec{\mathbf{x}})) - \frac{1}{k^2} \exp(H_k(\vec{\mathbf{x}})) \\ =& F(\vec{\mathbf{x}}) + G(\vec{\mathbf{x}}) - H(\vec{\mathbf{x}}) \end{aligned}$$

With each mixed-density B-spline function in $A(\vec{\mathbf{x}})$ given by:

$$f_j(x_j) = \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{f,(\rho,i,j)} S_{\rho,i}(x_j)$$
$$g_{k,j}(x_j)) = \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{g,(\rho,i,k,j)} S_{\rho,i}(x)$$
$$h_{k,j}(x_j)) = \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{h,(\rho,i,k,j)} S_{\rho,i}(x)$$

The norm of the gradient of $A(\vec{\mathbf{x}})$ with respect to trainable parameters is given by:

$$\left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} A(\vec{\mathbf{x}}) \right\|_1 = \left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} (F(\vec{\mathbf{x}}) + G(\vec{\mathbf{x}}) - H(\vec{\mathbf{x}})) \right\|_1$$
$$\left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} A(\vec{\mathbf{x}}) \right\|_1 = \left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} F(\vec{\mathbf{x}}) + \vec{\boldsymbol{\nabla}}_{\vec{\theta}} G(\vec{\mathbf{x}}) - \vec{\boldsymbol{\nabla}}_{\vec{\theta}} H(\vec{\mathbf{x}}) \right\|_1$$
$$\left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} A(\vec{\mathbf{x}}) \right\|_1 \leq \left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} F(\vec{\mathbf{x}}) \right\|_1 + \left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} G(\vec{\mathbf{x}}) \right\|_1 + \left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} H(\vec{\mathbf{x}}) \right\|_1$$

The first term is bounded,

$$\begin{aligned}
\left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} F(\vec{\mathbf{x}}) \right\|_1 &= \left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} \left(\sum_{j=1}^{n} f_j(x_j) \right) \right\|_1 \\
&= \left\| \sum_{j=1}^{n} \vec{\boldsymbol{\nabla}}_{\vec{\theta}} f_j(x_j) \right\|_1 \\
&\leq \sum_{j=1}^{n} \left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} f_j(x_j) \right\|_1
\end{aligned}$$

Substituting the expression for each $f_j(x_j)$, all lower densities have constant parameters:

$$\left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} F(\vec{\mathbf{x}}) \right\|_1 \leq \sum_{j=1}^{n} \left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} \left(\sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{f,(\rho,i,j)} S_{\rho,i}(x_j) \right) \right\|_1$$
$$\left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} F(\vec{\mathbf{x}}) \right\|_1 \leq \sum_{j=1}^{n} \left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} \left(\sum_{i=1}^{2^{r+2}} \theta_{f,(r,i,j)} S_{r,i}(x_j) \right) \right\|_1$$
$$\left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} F(\vec{\mathbf{x}}) \right\|_1 \leq \sum_{j=1}^{n} \sum_{i=1}^{2^{r+2}} \left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} \left(\theta_{f,(r,i,j)} S_{r,i}(x_j) \right) \right\|_1$$
$$\left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} F(\vec{\mathbf{x}}) \right\|_1 \leq \sum_{j=1}^{n} \sum_{i=1}^{2^{r+2}} |S_{r,i}(x_j)|$$

Each basis function is continuous and bounded by some positive constant $u > 0$, such that $S(x) < u$ regardless of its density, and it follows that:

$$\left\| \vec{\boldsymbol{\nabla}}_{\vec{\theta}} F(\vec{\mathbf{x}}) \right\|_1 \leq \sum_{j=1}^{n} \sum_{i=1}^{2^{r+2}} u$$

The last thing to include is that at most four basis functions are non-zero, regardless of the value of r, so a tighter upper bound is:

$$\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} F(\vec{\mathbf{x}})\right\|_1 \leq \sum_{j=1}^{n} \sum_{i=1}^{4} u = 4nu$$

Remark. Each ρ-density B-spline function has at most four active basis functions, and each mixed-density B-spline function has $r+1$ different ρ-density B-spline functions. If the lower densities $\rho < r$ were also trainable, then this upper bound would instead be $4nu(r+1)$. This is why only the maximum density was chosen to be trainable.

The exponential terms are more complicated.

$$\begin{aligned}
\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} G(\vec{\mathbf{x}})\right\|_1 &= \left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} \left(\sum_{k=1}^{M} \frac{1}{k^2} \exp(G_k(\vec{\mathbf{x}})) \right)\right\|_1 \\
\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} G(\vec{\mathbf{x}})\right\|_1 &\leq \sum_{k=1}^{M} \frac{1}{k^2} \left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} (\exp(G_k(\vec{\mathbf{x}})))\right\|_1 \\
\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} G(\vec{\mathbf{x}})\right\|_1 &\leq \sum_{k=1}^{M} \frac{1}{k^2} \left\|\exp(G_k(\vec{\mathbf{x}})) \vec{\boldsymbol{\nabla}}_{\vec{\theta}} (G_k(\vec{\mathbf{x}}))\right\|_1 \\
\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} G(\vec{\mathbf{x}})\right\|_1 &\leq \sum_{k=1}^{M} \frac{1}{k^2} \exp(G_k(\vec{\mathbf{x}})) \left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} (G_k(\vec{\mathbf{x}}))\right\|_1
\end{aligned}$$

Each mixed-density B-spline function is bounded, so

$$|g_{k,j}(x_j))| = \left| \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{g,(\rho,i,k,j)} S_{\rho,i}(x) \right| < u_{g,(k,j)}$$

Since n is fixed and finite, the functions $G_k(\vec{\mathbf{x}})$ are bounded:

$$|G_k(\vec{\mathbf{x}})| = \left| \sum_{j=1}^{n} g_{k,j}(x_j) \right| \leq \sum_{j=1}^{n} |g_{k,j}(x_j)| < \sum_{j=1}^{n} u_{g,(k,j)} = u_{g,(k)}$$

Since this is true for each G_k, one can choose the maximum bound:

$$u_g = \max_{k=1,\ldots,M} \{u_{g,(k)}\}$$

It is evident that:

$$G_k(\vec{\mathbf{x}}) \leq |G_k(\vec{\mathbf{x}})| < u_g$$

Since the exponential function is monotonic increasing:

$$\exp(G_k(\vec{\mathbf{x}})) \leq \exp(|G_k(\vec{\mathbf{x}})|) < \exp(u_g)$$

This result can be substituted back,

$$\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} G(\vec{\mathbf{x}})\right\|_1 < \sum_{k=1}^{M} \frac{1}{k^2} \exp(u_g) \left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} (G_k(\vec{\mathbf{x}}))\right\|_1$$

It should be noted that $G_k(\vec{\mathbf{x}})$ and $F(\vec{\mathbf{x}})$ have similar structure such that:

$$\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}}\left(G_k(\vec{\mathbf{x}})\right)\right\|_1 = \left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}}\left(F(\vec{\mathbf{x}})\right)\right\|_1$$

This is true, even though $G_k(\vec{\mathbf{x}}) \neq F(\vec{\mathbf{x}})$, because the same set of basis functions are used, with different coefficient parameters being the only difference. The consequence is that:

$$\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} G(\vec{\mathbf{x}})\right\|_1 < \sum_{k=1}^{M} \frac{1}{k^2} \exp(u_g) \left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}}\left(F(\vec{\mathbf{x}})\right)\right\|_1$$

Substituting previously shown results gives:

$$\begin{aligned}
\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} G(\vec{\mathbf{x}})\right\|_1 &< \sum_{k=1}^{M} \frac{1}{k^2} \exp(u_g) 4nu \\
\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} G(\vec{\mathbf{x}})\right\|_1 &< 4nu \exp(u_g) \sum_{k=1}^{\infty} \frac{1}{k^2} \\
\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} G(\vec{\mathbf{x}})\right\|_1 &< 4nu \exp(u_g) \frac{\pi^2}{6} < 4nu\pi^2 \exp(u_g)
\end{aligned}$$

The same argument can be used to find an upper bound for $\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} H(\vec{\mathbf{x}})\right\|_1$

$$\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} H(\vec{\mathbf{x}})\right\|_1 < 4nu \exp(u_h) \frac{\pi^2}{6} < 4nu\pi^2 \exp(u_h)$$

The original expression of interest was:

$$\begin{aligned}
\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} A(\vec{\mathbf{x}})\right\|_1 &\leq \left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} F(\vec{\mathbf{x}})\right\|_1 + \left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} G(\vec{\mathbf{x}})\right\|_1 + \left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} H(\vec{\mathbf{x}})\right\|_1 \\
\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} A(\vec{\mathbf{x}})\right\|_1 &< 4nu + 4nu\pi^2 \exp(u_g) + 4nu\pi^2 \exp(u_h) \\
\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} A(\vec{\mathbf{x}})\right\|_1 &< 4nu\pi^2 + 4nu\pi^2 \exp(u_g) + 4nu\pi^2 \exp(u_h) \\
\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} A(\vec{\mathbf{x}})\right\|_1 &< 4nu\pi^2 \left(1 + \exp(u_g) + \exp(u_h)\right)
\end{aligned}$$

Let the upper bound U be given by:

$$U = 4nu\pi^2 \left(1 + \exp(u_g) + \exp(u_h)\right)$$

From the definition of the norm $\|\|_1$, and the trainable parameters θ_i with index set Θ one has that:

$$\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} A(\vec{\mathbf{x}})\right\|_1 = \sum_{i \in \Theta} \left| \frac{\partial A}{\partial \theta_i}(\vec{\mathbf{x}}) \right|$$

Finally,

$$\left\|\vec{\boldsymbol{\nabla}}_{\vec{\theta}} A(\vec{\mathbf{x}})\right\|_1 = \sum_{i \in \Theta} \left| \frac{\partial A}{\partial \theta_i}(\vec{\mathbf{x}}) \right| < U$$

□

Remark. For a fixed number of variables n, the model has a total of $n2^{r+2}(2M+1)$ trainable parameters. The factor of k^{-2} inside the expression for Atlas is necessary to ensure the sum is convergent in the limit of infinitely many exponential terms $M \to \infty$. Only the maximum density ($\rho = r$) cubic B-spline function has trainable parameters, so that the gradient vector is bounded in the limit of arbitrarily large densities $r \to \infty$. It is worth recalling that at most four basis functions are active for uniform cubic B-spline functions, regardless of the density, but the smaller densities cannot be trainable, otherwise this property does not hold. The gradient vector has bounded norm for any number of basis functions and exponential terms. The bounded gradient vector implies that Atlas is numerically stable during training, regardless of its size or parameter count.

B.5.4 Atlas distal orthogonality

Property 3 (Distal orthogonality)**.** For any Atlas model $A(\vec{\mathbf{x}})$ and $\forall\ \vec{\mathbf{x}}, \vec{\mathbf{y}} \in D(A) \subset R^n$ and trainable parameters θ_i, there exists a $\delta > 0$ such that:

$$\min_{j=1,\ldots,n}\{|x_j - y_j|\} > \delta \implies \langle \vec{\boldsymbol{\nabla}}_{\vec{\theta}} A(\vec{\mathbf{x}}), \vec{\boldsymbol{\nabla}}_{\vec{\theta}} A(\vec{\mathbf{y}})\rangle = 0$$

Proof. Let $A(\vec{\mathbf{x}})$ denote some Atlas model, with mixed-density B-spline functions $f_j(x_j)$, $g_{i,j}(x_j)$, and $h_{i,j}(x_j)$ in the form:

$$\begin{aligned} A(\vec{\mathbf{x}}) =& \sum_{j=1}^{n} f_j(x_j) + \sum_{k=1}^{M} \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} g_{k,j}(x_j)\big) - \frac{1}{k^2} \exp\big(\Sigma_{j=1}^{n} h_{k,j}(x_j)\big) \\ =& F(\vec{\mathbf{x}}) + \sum_{k=1}^{M} \frac{1}{k^2} \exp(G_k(\vec{\mathbf{x}})) - \frac{1}{k^2} \exp(H_k(\vec{\mathbf{x}})) \\ =& F(\vec{\mathbf{x}}) + G(\vec{\mathbf{x}}) - H(\vec{\mathbf{x}}) \end{aligned}$$

With each mixed-density B-spline function in $A(\vec{\mathbf{x}})$ given by:

$$\begin{aligned} f_j(x_j) &= \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{f,(\rho,i,j)} S_{\rho,i}(x_j) \\ g_{k,j}(x_j)) &= \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{g,(\rho,i,k,j)} S_{\rho,i}(x_j) \\ h_{k,j}(x_j)) &= \sum_{\rho=0}^{r} \sum_{i=1}^{2^{\rho+2}} \theta_{h,(\rho,i,k,j)} S_{\rho,i}(x_j) \end{aligned}$$

Any mixed-density functions Φ and Ψ that act on different components of the input must have orthogonal parameter gradients, since each input variable has its own associated parameters:

$$\langle \vec{\boldsymbol{\nabla}}_{\vec{\theta}} \Phi(x_i), \vec{\boldsymbol{\nabla}}_{\vec{\theta}} \Psi(y_j)\rangle = 0 \ \forall\ i \neq j$$

Generally, since all mixed-density functions have parameters that are independent of each other it follows that for any mixed-density B-splines Φ and Ψ:

$$\langle \vec{\boldsymbol{\nabla}}_{\vec{\theta}} \Phi(x_j), \vec{\boldsymbol{\nabla}}_{\vec{\theta}} \Psi(y_j)\rangle = 0 \ \forall\ \Phi \neq \Psi$$

Thus, one need only compare the parameter gradients of each mixed-density B-spline function Ψ with itself. The inner-product of the parameter gradient of Ψ evaluated on two different inputs is given by:

$$\langle \vec{\boldsymbol{\nabla}}_{\vec{\theta}} \Psi(x_j), \vec{\boldsymbol{\nabla}}_{\vec{\theta}} \Psi(y_j)\rangle$$

The inner-product given above is not zero in general. However, as illustrated in Figure 18, for any mixed-density B-spline function Ψ there exist a $\delta > 0$, such that:

$$|x_j - y_j| > \delta \implies \langle \vec{\nabla}_{\vec{\theta}} \Psi(x_j), \vec{\nabla}_{\vec{\theta}} \Psi(y_j) \rangle = 0$$

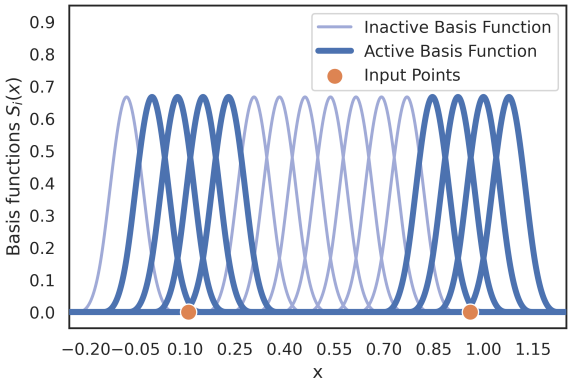

Figure 18: Visual proof of distal orthogonality for single-variable ρ-density B-splines.

This is because each basis function is zero everywhere, except on some small sub-interval. If this is true for all $j = 1, ..., n$, then the parameter gradients evaluated at $\vec{\mathbf{x}}$ and $\vec{\mathbf{y}}$ must be orthogonal. If this is true for all $j = 1, ..., n$, then it is true for the minimum. The converse is true by transitivity such that:

$$|x_j - y_j| > \delta \; \forall j = 1, ..., n \iff \min_{j=1,...,n} \{|x_j - y_j|\} > \delta$$

Finally,

$$\min_{j=1,...,n} \{|x_j - y_j|\} > \delta \implies \langle \vec{\nabla}_{\vec{\theta}} A(\vec{\mathbf{x}}), \vec{\nabla}_{\vec{\theta}} A(\vec{\mathbf{y}}) \rangle = 0$$

□

Remark. Two points that sufficiently differ in each input variable have orthogonal parameter gradients. It is worth mentioning that the condition resembles a cross-like region in two variables, and planes that intersect in higher dimensions. Distal orthogonality means Atlas is reasonably robust to catastrophic forgetting.