Supplement to ‘Budget Learning via Bracketing’

A Proofs Omitted from the Main Text

A.1 Proof of Theorem 1

Proof of lower bound. Notice that since $\mathcal{H}$ is one-sided learnable, it can learn any $h \in \mathcal{H}$ from below with $\mathsf{L} = 0$. Thus, given $m(\varepsilon, \delta, \lambda, \mathcal{H})$, and samples $(X_i, h(X_i))$ for any $h \in \mathcal{H}$, the scheme $\mathscr{A}$ recovers a function $\widehat{h}$ such that

$$\mu(h = 0, \widehat{h} = 1) \le \lambda$$
$$\mu(h = 0, \widehat{h} = 1) \le \varepsilon.$$

But then $\mu(\widehat{h} \neq h) \le \lambda + \varepsilon$ - i.e. $\mathscr{A}$ also serves as a realisable PAC learner with excess risk bounded by $\lambda + \varepsilon$. Thus, standard lower bounds for realisable PAC-learning can be invoked, for instance, that of §3.4 from the book Mohri et al. 2018. □

Proof of Upper Bound. We provide a scheme showing the same. To begin with, suppose that $\mathcal{H}$ is a finite class. Fix g, μ, and let $\mathcal{H}_\eta := \{h \in \mathcal{H} : \mu(g(X) = 0, h(X) = 1) \le \eta\}$. For finite $\mathcal{H}$, the scheme proceeds in two steps:

1. Testing: using m_1 samples (where m_1 is to be specified later), compute the empirical masses $\widehat{\ell}(h) := \widehat{\mu}\{h(X) = 1, g(X) = 0\}$ for every $h \in \mathcal{H}$. Let $\widehat{\mathcal{H}}_\lambda := \{h : \ell(h) < \lambda/2\}$.
2. Optimisation: Using m_2 samples (where m_2 is to be specified later), compute the empirical masses $\widehat{\mathsf{L}(h)} := \widehat{\mu}(h(X) = 0, g(X) = 1)$ for every $h \in \widehat{\mathcal{H}}_\lambda$. Return any $\widehat{h} \in \mathrm{argmin}_{\widehat{\mathcal{H}}_\lambda} \widehat{\mathsf{L}}(h)$.

The correctness of the above procedure is demonstrated by the following lemmata:

Lemma 4. *If*

$$m_1 \ge \frac{24}{\lambda} \log(4|\mathcal{H}|/\delta),$$

then with probability at least $1 - \delta/2$,

$$\mathcal{H}_{\lambda/4} \subset \widehat{\mathcal{H}}_\lambda \subset \mathcal{H}_{3\lambda/4}.$$

The above is proved after the conclusion of this argument.

Lemma 5. *If*

$$m_2 \ge \frac{2}{\varepsilon^2} \log(4|\mathcal{H}|/\delta),$$

then with probability at least $1 - \delta/2$,

$$|\widehat{\mathsf{L}(h)} - \mu(h = 0, g = 1)| \le \varepsilon$$

simultaneously for all $h \in \widehat{\mathcal{H}}_\lambda$.

Proof. The claim follows by Hoeffding’s inequality and the union bound, noting that $|\widehat{\mathcal{H}}| \le |\mathcal{H}|$. □

Thus, for finite classes, the claim follows (with $d = \log|\mathcal{H}|$) by an application of the union bound, and noting that $\mathcal{H}_0 = \{h : \mu(h = 1, g = 0) = 0\} \subset \{h : \mu(h = 1, g = 0) \le \lambda/4\} = \mathcal{H}_{\lambda/4}$.

We now appeal to the standard generalisation from finite classes to finite VC-dimension classes. By the Sauer-Shelah lemma (see, e.g., §3.3 of Mohri et al. 2018), with m samples, a class of VC-dimension d breaks into at most $(em/d)^d$ equivalence classes of functions that agree on all data points, and the losses of functions in each equivalence class can be simultaneously evaluated and share the same generalisation guarantees. Let $\mathcal{H}'$ be formed by selecting one representative from each such class. We may run the above procedure for $\mathcal{H}'$, and draw the same conclusions so long as

$$m \ge m_1 + m_2$$
$$m_1 \ge \frac{24}{\lambda} (d \log(em/d) + \log(4/\delta))$$
$$m_2 \ge \frac{1}{2\varepsilon^2} (d \log(em/d) + \log(4/\delta))$$

By crudely upper bounding the right hand sides above, this can be attained if

$$\frac{m}{\log em} \geq 24\left(\frac{1}{\lambda}+\frac{1}{\varepsilon^2}\right)(d+\log(4/\delta)),$$

and the conclusion follows on noting that for $v \geq 2$, $u \geq 4v\log(v) \implies u/\log(eu) \geq v$. □

It remains to show Lemma 4.

Proof of Lemma 4. Let $\ell(h) := \mu(h=1, g=0)$. Note that $m_1\widehat{\ell(h)}$ is a Binomial$(m_1, \ell(h))$ random variable for each h. Further, for $p \leq q$, the distribution Binomial(n,q) stochastically dominates Binomial(n,p).

Thus, for any $h : \ell(h) < \lambda/4$,

$$\mu^{\otimes m_1}(\widehat{\ell}(h) \geq \lambda/2) \leq P_{U\sim\text{Binomial}(m_1,\lambda/4)}(U \geq m_1\lambda/2) \leq \exp(-3m_1\lambda/32),$$

where the final relation is due to Bernstein's inequality.

Similarly, for any $h : \ell(h) > 3\lambda/4$,

$$\mu^{\otimes m_1}(\widehat{\ell(h)} \leq \lambda/2) \leq P_{U\sim\text{Binomial}(m_1,3\lambda/4)}(U \leq m_1\lambda/2) \leq \exp(-m_1\lambda/24).$$

For $m_1 \geq 24/\lambda\log(4|\mathcal{H}|/\delta)$, each of the above can be further bounded by $\delta/4|\mathcal{H}|$. The claim follows by the union bound. □

A.1.1 Alternate Generalisation Analyses

Note that the above proof utilises the finite VC property only to assert that on a finite sample, the hypotheses to be considered can be reduced to a finite number. Instead of the VC theoretic argument, one can then immediately give analyses via, say, L_1 covering numbers of the sets induced by the functions. Similarly, instead of beginning with finite hypotheses, we may instead directly uniformly control the generalisation error of the estimates for each function via the Rademacher complexity of the class $\mathcal{H}$, thus replacing Lemmas 5, 4 by a bound of the form $m(\varepsilon,\lambda,\delta,\mathcal{H}) \leq \inf\{m : \mathfrak{R}_m(\mathcal{H}) + \sqrt{2\log(2/\delta)/m} \leq \min(\lambda,\varepsilon)/2\}$, and further extensions via empirical Rademacher complexity. In addition, one can utilise more sophisticated analyses for more sophisticated algorithms.

The point of all this is to underscore that once one adopts the bracketing and OSL setup, generalisation guarantees, and thus sample complexity bounds, follow the standard approaches in learning theory. This is not to say that these analyses may be trivial - for instance, in the above we have not shown tight sample complexity bounds at all.

A.2 Proof of Theorem 2

Proof of Upper Bound. We note that if $\frac{\mathrm{d}\mu}{\mathrm{dVol}} \geq \rho$, and we can locally predict in a region of volume P, then we can immediately locally predict in a region of μ-mass ρP. Thus, it suffices to argue the claim for the Lebesgue mass on $[0,1]^p$.

Since we have access to κ cuboids in $\mathcal{R}^{0,1}_\kappa$, we can capture any κ of the cuboids induced in the minimal partition aligned with g for any $g \in \mathcal{G}$. In particular, when approximating from below, we will choose h^- to be 1 on some κ of the cuboids contained in $\{g=1\}$, and 0 otherwise, and similarly for approximating from above (denoted h^+). Naturally, we will 'capture' the cuboids with the biggest volume (more generally, biggest μ mass). Notice that this construction trivially yields $h^- \leq g \leq h^+$.

To finish the argument, fix an arbitrary $g \in \mathcal{G}$. Let $\mathscr{P}$ be a partition aligned with g that is V-regular, and further, has the largest total number of parts possible.[11] Suppose that there are $\mathscr{P}_1$ parts in $\mathscr{P}$ on which g is 1, and $\mathscr{P}_0$ on which it is 0. By the maximality, it must be the case that each rectangle contained in each part of $\mathscr{P}$ has volume less than $2V$, since otherwise we can split this part while maintaining V-regularity. Further, since the

[11] such a partition exists because V-regularity implies that the number of parts is uniformly bounded by $1/V$.

mass contained outside of the rectangle in each part is at most V, it follows that $3V(\mathscr{P}_0 + \mathscr{P}_1) \geq 1$ by the union bound. Thus, $\mathscr{P}_0 + \mathscr{P}_1 \geq 1/3V \geq \kappa/3$.

Now, by the above construction, we can capture at least $(\min(\kappa, \mathscr{P}_0) + \min(\kappa, \mathscr{P}_1))V$ volume of the space, which exceeds $\kappa V/3$. □

Proof of Lower Bound. Divide $[0,1]^p$ into $N = \lfloor 1/V \rfloor$ congruent, disjoint rectangles. Note that since the faces of these rectangles have codimension ≥ 1, they have volume 0. Thus, we need not worry about how they are assigned in the following, and we will omit these irrelevant details in the interest of clarity.

We set $\mathcal{G}$ to be the class of 2^N functions obtained by colouring each of the N boxes as 0 or 1. This class is trivially V-regular.

Now, notice that any time a function h is approximating a function $g \in \mathcal{G}$ from above, it should either attain the value 0 on a whole box, or attain the value 1 on a whole box - if g is 1 on a box, then h is forced to be 1. If g is instead 0, and h dips down to take the value 0 at any point, then rising up to 1 is lossy in that it increases the loss $\mathsf{L}(h, g, \mathrm{Vol})$ while offering no reduction in the expressivity of the class $\mathcal{H}$. Thus, we may restrict attention to classes $\mathcal{H}$ such that all functions contained in them are constant over the boxes described.

Given the above setup, the entire problem is equivalently described by restricting the domains of $\mathcal{G}, \mathcal{H}$ to the centres of the above boxes, and the measure Vol to the uniform measure over these centres. We henceforth work in this space. The domain of the functions in $\mathcal{G}, \mathcal{H}$ is now the abstract set $[1:N]$.

Suppose every $g \in \mathcal{G}$ can be budget learned with budget at most $1 - \Delta/N$ in this measure (where Δ is some integer because the space is discrete and the distribution is rational). Let (h_g^+, h_g^-) be the appropriate bracketing functions that minimise budget for g, and let $\mathcal{I}_g$ be the points where $h_g^+ = h_g^-$. The budget constraint forces that $|\mathcal{I}_g| \geq \Delta$. Notice that outside of $\mathcal{I}_g$, h_g^+ must take the value 1 and h_g^- must take the value 0 - indeed, if $h_g^+(i)$ was 0, then since $0 \leq h_g^-(i) \leq h_g^+(i)$, $h_g^-(i) = 0$, and then $i \in \mathcal{I}_g$.

But, on $[1:N] \sim \mathcal{I}_g$, g must either be predominantly 1 or 0, and then respectively, must agree with h_g^+ or h_g^- on at least $(N - |\mathcal{I}_g|)/2$ points. This means that there exists a $h'_g \in \mathcal{H}$ (which is either h_g^+ or h_g^-) such that

$$|\{i : h'_g(i) = g(i)\} \geq |\mathcal{I}_g| + \frac{N - |\mathcal{I}_g|}{2} \geq \frac{N + \Delta}{2}.$$

With this setup, we invoke the following statement

Lemma 6. *If a class of functions $\mathcal{F}$ on $[1:N]$ of VC-dimension $d \leq N$ is such for every $\{0,1\}$-valued function on $[1:N]$, there exists a $f \in \mathcal{F}$ that agrees with it on at least $(N+\Delta)/2$ points, then the VC-dimension of $\mathcal{F}$ is at least $\frac{3\Delta^2}{2(N+\Delta)\log(eN)} \geq \frac{3\Delta^2}{4N\log(eN)}$.*

Notice that since $N \geq \Delta$, Invoking the above, and the fact that the VC-dimension of $\mathcal{H}$ is at most d, it follows that (for $N \geq 3$)

$$\frac{3\Delta^2}{8N\log N} \leq d \iff \Delta \leq \sqrt{3dN\log N},$$

from which the claim is immediate on recalling that $1/V \geq N \geq 1/V - 1$. □

Proof of Lemma 6. Identify all $\{0,1\}$ labellings as above with the cube $\{0,1\}^N$, and similarly the patterns achieved by $\mathcal{F}$ as a subset of the same. The hypothesis is then equivalent to saying that for every point $p \in \{0,1\}^N$, there exists a point $f \in \mathcal{F}$ such that $d_{\mathrm{H}}(p, f) \leq \frac{N-\Delta}{2}$, were d_{H} is the Hamming distance. But then $\mathcal{F}$ is a $(N-\Delta)/2$-cover of the Boolean hypercube.

By a standard volume argument, it then must hold that

$$|\mathcal{F}| \geq \frac{2^N}{\sum_{i=0}^{(N-\Delta)/2} \binom{N}{i}} \geq e^{+\frac{3}{2}\frac{\Delta^2}{N+\Delta}}$$

where the final inequality follows on noting that the right hand side of the first inequality is 1 divided by a lower tail probability for N independent fair coin flips, and then invoking Bernstein's inequality.

However, by the Sauer-Shelah Lemma, if $d \le N$ is the VC-dimension of $\mathcal{F}$, then the number of elements in it is at most

$$\sum_{i=0}^{d} \binom{N}{i} \le \left(\frac{eN}{d}\right)^{d}.$$

Relating these, we have

$$e^{+\frac{3}{2}\frac{\Delta^2}{N+\Delta}} \le (eN/d)^d \iff \frac{3\Delta^2}{2(N+\Delta)\log(eN/d)} \le d. \qquad \square$$

A.3 Proof of Theorem 3

These lower bounds are proved similarly to the lower bound from the previous section: principally, they use the fact that any non-trivial budget learner also yields non-trivial coverings, and construct function classes of limited VC dimension with large covering numbers.

Proof of the bound (i). Let $S := \{x_1, \dots, x_D\}$ be a set of shattered points. The measure μ_S is set to the uniform distribution on S. The restriction $\mathcal{G}_{|S}$ consists of all $\{0,1\}$-valued functions on D points. If $\mathcal{H}$ can budget learn this with respect to μ_S with budget $1 - \Delta/D$, then $\mathcal{H}_{|S}$ is a $(D-\Delta)/2$covering of $\{0,1\}^S$. Invoking Lemma 6 just as in the proof of the lower bound in the previous section, we get that $\frac{\Delta}{D} \ge \sqrt{3\frac{\mathrm{VC}(\mathcal{H})}{D}\log(\frac{eD}{\mathrm{VC}(\mathcal{H})})}$. □

Proof of the bound (ii). We use a class on $[1:N]$, constructed by Haussler 1995 that is known to have large packing number. Note that the same class is used as an example of a simple budget-learnable class in §3.4. The class is defined as follows: Suppose D divides N. Let $\mathcal{F}$ be the class of single thresholds on $[1:N/D]$, i.e.$\mathcal{F} = \{f_k, k \in [0:N/D+1]\}$, where $f_k(i) := \mathbb{1}\{k \le i\}$. $\mathcal{F}$ trivially has a VC-dimension of 1. $\mathcal{G}$ is generated as a tensor product of D copies of $\mathcal{F}$ placed on a partition of $[1:N]$. Concretely, we may say that each $g \in \mathcal{G}$ can be represented as D functions $(f_{k_1}, f_{k_2}, \dots, f_{k_D}) \in \mathcal{F}^{\otimes D}$ for some $k_1, \dots k_d \in [0:N/D+1]$ such that for $i \in [jN/D+1:(j+1)N/D]$ for any $j \in [0:D-1]$, $g(i) = f_{k_j}(i)$.

Haussler 1995 shows that for this class, under the uniform measure on $[1:N]$, the k-packing number is at least $\frac{(1+N/D)^D}{2^D\binom{k+D}{D}}$. Now recall that the $k/2$-covering number must exceed the k-packing number for any set and metric. Further, a budget of $1-\Delta/N$ implies a $\frac{N-\Delta}{2}$-covering. The budget requirement imposes the condition $N - \Delta \le \mathsf{B}N$. Thus, invoking Sauer-Shelah as in the proof of Lemma 6, we obtain

$$\begin{aligned}\left(\frac{eN}{d}\right)^d &\ge \left(\frac{N+D}{2e(N+D-\Delta)}\right)^D \\ &\ge \left(\frac{N+D}{2e(\mathsf{B}N+D)}\right)^D \\ &\ge \left(\frac{1}{4e\mathsf{B}}\right)^D\end{aligned}$$

where we have used that $\mathsf{B}N \ge D$ in the final line. The above bound is non-vacuous only if $4e\mathsf{B} < 1$.

The case $\mathsf{B} < D/N$ is not discussed in the theorem, since it is a vanishingly small budget, but by the above, in this case we get a lower bound of $(N/4eD)^D$ in the above, giving, for $D \lesssim N^{1-\epsilon}$ for some $\epsilon > 0$, a bound of $d = \Omega(D)$ in this setting. □

A.4 Proofs of budget claims made in §3.4

Proof for sparse VC classes. fix any g. We pick the function that is 1 on the d choices of $i \in g^{-1}(1)$ with the largest total μ-mass as the lower approximation, and the constant 1 as the approximation from above. □

Proof for Tensorised class. The class naturally breaks the domain into D equal parts, and places a threshold on each. We choose the $d-1$ parts with largest μ-mass, and place a threshold there. Lastly, we collate the remaining parts into one set, and we place the constant functions 1 and 0 on this. A tensorisation of these function classes demonstrates the claim. □

Proof for Convex Polygons. Instead of approximation from above and from below, we will adopt the more natural terminology of inner and outer approximation. As the class is closed under $f \mapsto 1-f$, to show budget learnability with budget B, it suffices to show that for any polygon P with D vertices and any measure μ, there exist polygons $P_{\text{in}} \subset P \subset P_{\text{out}}$ of d vertices such that $\mu(P_{\text{in}}) \ge (1-\mathsf{B})\mu(P)$ and $\mu(P^c_{\text{out}}) \ge (1-\mathsf{B})\mu(P^c)$. This follows since the cloud query points are precisely those in $P_{\text{out}}/P_{\text{in}})$, which has mass $\mu(P_{\text{out}}) - \mu(P_{\text{in}}) \le 1 - (1-\mathsf{B})(1-\mu(P)) - (1-\mathsf{B})\mu(P) = 1 - (1-\mathsf{B}) = \mathsf{B}$.

Inner Approximation: We offer a direct proof. Consecutively number the vertices of P as $[1:D]$. Form the d-gon P^1 using the vertices $[1:d]$. Remove this polygon from P and relabel $1 \mapsto 1, d \mapsto 2, \dots, n \mapsto n+2-d, \dots$. Contuining this process $m := \lceil D/d-2 \rceil$ times partitions P into m d-gons $P^1, \dots, P^m$. By the union bound, $\sum \mu(P^i) \ge \mu(P)$. But then there must exist at least one d-gon $P_{\text{in}} \subset P$ such that $\mu(P_{\text{in}}) \ge \mu(P) \ge \frac{1}{\lceil D/d-2 \rceil}\mu(P)$.

Outer Approximation: Recall that $d \ge 4$, and $D \ge d$. We will show that for any D-gon P there exists a d-gon P_{out} containing it such that $\mu(P^c_{\text{out}}) \ge \frac{d-2}{D-2}\mu(P^c)$.

We induct on D. As a base case, for $D = d$, the claim holds trivially since P itself may serve. Let us assume the claim for D-gons, and let P be a $D+1$-gon. Note that since $D \ge 4, D+1 \ge 5$. Thus, P has at most two pairs of consecutive exterior angles that are each exactly $\pi/2$ (since the sum of all exterior angles is 2π, and P has at least 5 exterior angles). For any side such that the two exterior angles are not both $\pi/2$, the sides preceding and following it (in the cyclic order) may be extended to meet at some point. This yields a triangle with this side as a base. Since such an extension can be done for at least $D+1-2 = D-1$ sides, this yields $D+1 \ge J \ge D-1$ triangles $\triangle_1, \triangle_2, \dots, \triangle_J$. Now notice that for each $j \le J$, $Q_j := P \cup \triangle_j \supset P$ is a D-gon. Further, by the union bound, $\sum \mu(\triangle_j) \le \mu(P^c)$, and thus there exists a triangle $\triangle_{i^*}$ such that $\mu(\triangle_{i^*}) \le \mu(P^c)/J$, and thus $\mu(Q^c_{i^*}) \ge \frac{J-1}{J}\mu(P^c) \ge \frac{D-2}{D-1}\mu(P^c)$. Now, by the induction hypothesis, there exists a d-gon P_{out} containing Q_{i^*} (and hence P) such that $\mu(P^c_{\text{out}}) \ge \frac{d-2}{D-2}\mu(Q^c_{i^*}) \ge \frac{d-2}{D-1}\mu(P^c)$. This concludes the argument.

Thus, we can attain the budget

$$\mathsf{B} = 1 - \min\left(\frac{1}{\lceil \frac{D}{d-2} \rceil}, \frac{d-2}{D-2}\right) = 1 - \left\lceil \frac{D}{d-2} \right\rceil^{-1}. \qquad \square$$

B Experiments

B.1 Losses and algorithms for methods listed in §4

We list the general approach taken for each of the methods we compare to. More precise details very between datasets, and are described in subsequent sections. Note that all models are trained on GPUs using stochastic gradient descent for linear models and ADAM for deep networks. In each case, a multitude of models are trained by scanning over values for the relevant Lagrange multiplier/regularisation weight. The collection of models so obtained is tuned, and then a model finally selected for each target accuracy via procedures detailed in §B.5.

Bracketing The general approach, and a formulation for generic loss functions is given in (1) in §1.3. The exact loss formulation used in the experiments is the following,

$$\hat{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} -1_{g(x_i)=1}\log\left(h_\theta(x_i)\right) - \xi 1_{g(x_i)=0}\log\left(1 - h_\theta(x_i)\right) \tag{2}$$

where ξ is a hyper parameter between two components of loss function. The term multiplying ξ is the constraint, which imposes a high cost in case of a leakage. The other term in the loss objective pushes the model to increase true positives. For example, if ξ is 0, local model always predicts 1 and it has maximum leakage and minimum budget. If ξ is $+\infty$, the local model always predicts 0 and it has minimum leakage and maximum budget.

Local Thresholding We first train a local predictor using the cross entropy loss and freeze it. We rank the examples based on maximum of the prediction probabilities. We select a threshold and the predictor uses cloud model if its current maximum probability is lower than threshold. We attain different budget values by changing this threshold.

Alternating Minimisation (Nan and Saligrama 2017a) we follow the ADAPT-LIN procedure from this paper, which is an alternative minimisation scheme between an auxiliary q and local predictors & gating. Since we don't have feature costs in our setting, we assumed $\gamma = 0$ in our experiments. We stopped the procedure if the q vector converges, or if a predefined number of iterations - in our case 10 - is exceeded. Different budget values are obtained by sweeping values of the regularisation parameter - in this paper called λ.

Sum relaxation (Cortes et al. 2016) utilising the relaxation as developed in this paper, we use the loss $L_{MH}(h, r, x, y)$ formulated within as a loss function to train a neural network. This is optimised with several values of the regularisation parameter, c, to obtain different usage values.

Selective Net (Geifman and El-Yaniv 2019) we follow the architectural augmentations and losses as prescribed by this paper. We train the network with auxiliary head and ignore this part during inference time. Again, this is performed for several values of the Lagrange multiplier, called c here as well.

B.2 Synthetic Data

Cloud Classifier A training dataset of 2.5K points was sampled uniformly from the set $[-10, 10] \times [-10, 10]$. The complex classifier's decision boundary can be expressed as

$$\mathbb{1}\left\{x + 4x^2 + 3x^3 + 3x^4 + y + y^2 + y^3 + y^4 + 5xy^2 + 30x^2y < 1000\right\}$$

where x, y are the coordinates of the data point.

Local Classifier Weak learners are restricted to axis-aligned conic sections, which may be implemented as linear classifiers which see input features x, y, x^2, y^2.

Training Details Each weak learner model has hyper parameters which are adjusted to observe the power of the methods. As an example, learning rates are chosen in the range of $[10^{-5}, 10^{-2}]$, ξ value for bracketing model is chosen in the range of $[1, 3]$, λ values for alternating minimisation are chosen in the range of $[0.25, 0.75]$ and c values for the sum relaxation method are chosen in the range of $[0, .3]$. After obtaining several models, the best models are reported based on the true error rates and true usages.

B.3 MNIST Odd/Even

Cloud Classifier We implement a LeNet architecture with 6 filters in the first convolution layer, 16 filters in the second convolution layer, 120 neurons in the first fully connected layer and 84 neurons in the first fully connected layer. Kernel size for convolution layers is chosen to be 5. Overall, this model has $43.7K$ parameters. Learning rate is chosen to be 10^{-3} and it is halved in every 20 epochs for a total of 60 epochs using 64 as batch size. L_2 regularisation of 10^{-5} is applied. The model attains 99.46% test accuracy.

Local Classifier Linear classifiers are adopted as weak learner architecture - these have $1.57K$ parameters, and no convolutional structure. Half of the training set $(30K)$ is randomly chosen to be weak learner dataset. Within this dataset, 90% $(27K)$ is kept as training set for and 10% $(3K)$ as validation. Training and validation sets for each of the methods are kept the same to ensure a fair comparison. The local model attains 89.79% test accuracy.

Training Details For each model, learning rate is chosen to be 10^{-2} and it is halved in every 25 epochs for a total of 120 epochs. Batch size is chosen to be 64 and L_2 regularisation of 10^{-5} is applied. For bracketing, ξ values are chosen in the range of $[0, 24]$ for a total of 21 values. For alternating minimisation, λ values are swept in the range $[0, 1]$ for a total of 25 values and a maximum of 10 alternative minimisation rounds are allowed. For the sum relaxation, c is chosen in the range $[0, .495]$ for a total of 25 values. For the selective net, c values are chosen in range $[0, 1]$ for a total of 25 values. We note here that the auxiliary head in the selective net, which serves in deep networks as a way to improve feature extraction, is ineffective in this linear setting.

B.4 CIFAR Random Pair

Cloud Classifier We pick ResNet32 (He et al. 2016) as the high-powered model and trained it, with configurations as described by Idelbayev 2019, on the full multi-class CIFAR training data. This model has $.46M$

parameters.

Local Classifiers We pick a narrow LeNet model as weak learner that has 3 filters in the first and second convolution layers, and 15 neurons in the first fully connected layer. Kernel size for convolution layers is chosen to be 5. Overall, this weak model has $1,628$ parameters.

Procedure for training For each run, we choose 2 classes out of 10 CIFAR classes randomly and extract the subset of the dataset corresponding to this couple. The cloud classifier is obtained using the pre-trained ResNet32 and only retraining the prediction layer while keeping the backbone frozen for this binary dataset. Learning rate is chosen to be 10^{-2} and it is halved after 50 epochs for a total of 100 epochs. Batch size is chosen to be 64 and L_2 regularisation of 10^{-5} is applied. The model attains on average 98.38% test accuracy.

For the weak learners, 60% (6K points) of the training set is randomly chosen to be the training dataset. From this, 83.3% (5K) is kept as training set for and 16.7% (1K) culled for validation. The model attains on average 90.94% test accuracy. Training and validation set are kept the same across methods to have a fair comparison.

Training Details Learning rate is chosen to be 10^{-3} and it is halved in every 75 epochs for a total of 300 epochs. Batch size is chosen to be 64 and L_2 regularisation of 10^{-5} is applied. For bracketing model, values for ξ are chosen in the range of $[0, 65]$ for a total of 36 values. For alternating minimisation λs are swept in the range $[0, 1]$ for a total of 40 values and a maximum of 10 alternative minimisation rounds are allowed. For the sum relaxation method c values are chosen in range $[0, .495]$. For each of the above methods, all the networks are warm started using the parameters of the local model. Note each of the previous methods implement two Narrow LeNets - for bracketing these are the two one-sided learners, while for the other two, these are gates and predictors. For the selective net, c values are chosen in range $[0, 1]$ for a total of 40 values. Warm starting this network leads to lowered performance than random initialisation, and so the latter values are reported.

The above procedure is performed for 10 trials of random classes of CIFAR. These classes are listed in Table 4 below, along with usages attained for the bracketing and selective net methods in these cases. Only these two methods are reported here since they are the most competitive of the five.

B.5 Model Selection Process

This section explains the model selection procedure for the methods.

For each value of the Lagrange multiplier/regularisation constant chosen in the above training methods, we receive a model (or a pair of models, as appropriate). Let this collection of models be $\mathcal{M}$. These models have real valued outputs in the range $[0, 1]$, and a decision needs to be extracted from these. In order to provide sufficient granularity to the models that they be able to match any required target accuracy, we vary the threshold of output value at which the models' decisions go from 0 to 1. This process differs in details for different methods. The tuning is performed

Local Thresholding In this case $\mathcal{M}$ is a singleton. We compute the cross entropy of the classifier's output and abstain if this cross entropy is larger than a threshold τ that is selected as follows: the values of τ considered are obtained by computing the cross entropies of the model outputs on each of the training points. On validation data, usages and accuracy are computed for the models which thresholds at each of the considered thresholds. At a given target accuracy, the value of τ which yields at least this accuracy on the validation data with the smallest usage is selected.

Bracketing Note that each $m \in \mathcal{M}_{\text{bracketing}}$ contains two models $(m_{\text{below}}, m_{\text{above}})$ which are respectively approximations from above and below - these may be trained with different ξ, thus giving a total of $|\Xi|^2$ models. Suppose the target accuracy is $1 - \alpha$. Let the training data have size T. Using the training data, for every $i \in [0 : \alpha T]$, we determine pairs of thresholds $\tau_m(i) = (\tau^m_{\text{below}}(i), \tau^m_{\text{above}}(i))$ such that the leakages of $(m_{\text{below}}, m_{\text{above}})$ on the training data are exactly i/T each. This then gives us a total of at most $|\Xi|^2 \times \alpha T$ possible model-threshold pairs, represented as $(m, \tau_m(i))$.

Now, each of these tuples is evaluated on the validation data, with usages and accuracies computed. Again, the pair of models and thresholds with the smallest usage that exceeds the target accuracy on the validation set is selected.

Alternating Minimisation *and* Sum Relaxation *and* Selective Net Each $m \in \mathcal{M}$ is a pair (γ, π), where the former is the gate. Again, on the training data, the value taken by γ on each training point is recorded. This gives all the thresholds that may be selected for the gating function. Now, each m and corresponding choice of threshold may be evaluated on the validation set, and we select the ones which match the accuracy requriement and show the lowest usage.

B.6 Tables Omitted from the Main Text

Task	**Target**	**Bracketing**			**Local Thr.**			**Alt. Min.**			**Sum relax.**			**Sel. Net.**			**Gain**
	Acc.	**Acc.**	**Usg.**	**ROL**	**Acc.**	**Usg.**	**ROL**	**Acc.**	**Usg.**	**ROL**	**Acc.**	**Usg.**	**ROL**	**Acc.**	**Usg.**	**ROL**	
MNIST Odd/Even	0.995	0.994	0.457	2.19	0.995	0.653	1.53	0.991	0.830	1.20	0.997	0.785	1.27	0.996	0.658	1.52	1.431×
	0.990	0.990	0.387	2.58	0.991	0.515	1.94	0.985	0.740	1.35	0.992	0.651	1.54	0.992	0.544	1.84	1.332×
	0.980	0.982	0.299	3.35	0.983	0.358	2.79	0.974	0.604	1.66	0.992	0.651	1.54	0.985	0.423	2.37	1.199×
CIFAR Random Pair	0.995	0.991	0.363	4.01	0.996	0.510	2.25	0.991	0.854	1.19	0.997	0.620	2.07	0.992	0.436	3.04	1.280×
	0.990	0.986	0.294	5.66	0.991	0.399	3.41	0.986	0.754	1.40	0.994	0.488	3.31	0.987	0.347	4.30	1.265×
	0.980	0.975	0.214	9.97	0.983	0.276	6.38	0.975	0.611	1.87	0.986	0.345	5.81	0.977	0.257	11.67	1.195×

Table 3: Performances on BL tasks studied. This table repeats the entries of Table 1, with the addition of a column indicating the test accuracy attained by the models. Note that these fluctuate in the range -0.05 to +0.02 of the target, as can be expected from any selection method.

Class Pair	**Bracketing**	**Sel. Net.**	**Gain**
0 - 3	0.304	0.364	1.199×
6 - 4	0.452	0.526	1.164×
5 - 2	0.616	0.631	1.026×
6 - 1	0.095	0.122	1.296×
9 - 3	0.220	0.211	0.961×
8 - 1	0.235	0.381	1.619×
7 - 4	0.615	0.646	1.050×
8 - 7	0.059	0.091	1.538×
4 - 0	0.195	0.315	1.620×
6 - 7	0.152	0.179	1.177×

Table 4: Usages and relative gain for bracketing and selective net (Geifman and El-Yaniv 2019) method for 99% target accuracy for 10 CIFAR random pairs. These two methods uniformly have the lowest usages, and hence the others are omitted. All models achieve test accuracy in the range 98.1-99.3% test accuracy.