A. Inductive Bias

A.1. Generalizing known choice models

As stated in Claim 1, the model in Eq. 5 generalizes many models that have been proposed in the discrete choice literature. Table 2 summarizes the specific instantiations of $w, r$ and $x_i$ for these models, partitioning according to whether they incorporate set dependent weights, set dependent representations, or both.

A.2. Illustrative example of aggregation principles

Section 2.1 described four principles from behavioral choice theory that have guided our model design choices. Here we give a short illustrative example with $\ell = 2$ of how these principles come into play. Consider a user choosing an item from a set of alternatives. The first principle states that item values are considered along multiple dimensions. These could correspond to explicit item features (e.g., price, size), but in many cases are latent, and in our framework, learning the $f_1$ amounts to inferring these latent dimensions. For our example, assume the model has learned two such dimensions with $f_1$ and $f_2$, and that for example’s sake, these correspond to notions of perceived “cost” and “quality”, respectively.

The second principle states that within each dimension, value is relative, and is considered in relation to a set-dependent reference point. In our example, this would mean that users value the cost of an item not by it’s absolute value under $f_1$, but rather, by it’s value under $f_1$ relative to a reference point $r(\hat{s}_1)$ (and similarly for $f_2$). For example, if $r$ is set to the average cost (i.e., average value of $f_1$) for the set, then cost is perceived relative to this baseline via $x_i - r(\hat{s}_1)$.

Given the above, note that items valued higher than the reference value are perceived as “gains”, and items valued lower are perceived as “losses”. The third principle (central to prospect theory) states that the perception of losses and gains follows an a-symmetric s-shaped curve, meaning that losses loom greater (negatively) than gains do (positively). In our model, this role is played by $\mu$. In our example, this would mean that items with below-average quality would be perceived as “losses”, items with above-average quality would be perceived as “gains”, and that low quality “hurts” value more than high quality “helps” it.

The forth and final principle states the degree to which each valuation dimension contributes to the overall perceived value is also set-dependent, or in other words, the choice set also determines which valuation dimensions are important. In our example, this could mean that if perhaps the items are similar in terms of cost, then the importance of quality would be amplified, and vice versa.

B. Approximation Error

In this section we provide the proof for Theorem 1. Before giving the proof, we highlight some definitions and results from Ambrus & Rozen (2015) that we will use, and define the base class we consider.

B.1. Results from Ambrus & Rozen (2015)

Ambrus & Rozen (2015) study aggregation from a different perspective than ours, but provide a theoretical foundation for reasoning about aggregators public to use in our proof. The setting and tasks considered in Ambrus & Rozen (2015) are quite different from ours. Specifically, they focus on a realizable, worst-case, non-parametric setting: there is a fixed and finite grand set of (non-featurized) items, choices are set by a deterministic choice function $y = c(s)$, items are scored by item-wise utility functions mapping items to arbitrary values (i.e., there is no notion of function class structure or complexity), and the goal is to fully reconstruct $c$ (i.e., matching its predictions on all possible choice sets) by aggregating utility functions. The main results in their paper quantify the number of utility functions necessary for full reconstruction under certain conditions and as a function of the “number of violations” of IIA (which they define). In contrast, we consider aggregation from a statistical perspective, focusing on a setting that is typical in machine learning: items are featurized, choice sets and choices are drawn i.i.d. from an unknown joint distribution, score functions are parametric, and the goal is to learn a predictor with high expected accuracy.

Since Ambrus & Rozen (2015) work with a finite set of $N$ items, item utilities (which in our case would be modeled using item-wise functions) are expressed as vectors $u \in \mathbb{R}^N$ with entries corresponding to the utilities of items. For a collection of utilities $u = \{u_1, \ldots, u_\nu\}$ where each $u_i \in \mathbb{R}^N$, we slightly abuse notation and use $g_u$ to denote an aggregator with item-wise score functions corresponding to the utilities in $u$.

A key insight of Ambrus & Rozen (2015) is that to reason about aggregation with $N$ items, it suffices to consider the behavior of an aggregator on an arbitrary set of three items $x_1, x_2, x_3$. The following definition of a triple basis (TB) constitutes the main building block of Ambrus & Rozen (2015).

Definition 3 (Triple basis, Ambrus & Rozen (2015)). Let $x_1, x_2, x_3 \in N$. Let $\nu \in \mathbb{N}$, and let $u = \{u_1, \ldots, u_\nu\}, u_i \in \mathbb{R}^N$. Then $u$ is a triple basis for $x_1, x_2, x_3$ under the aggregation mechanism $\psi$ if:

1. $x_1$ is strongly preferred to $x_2$ from the choice set $s = \{x_1, x_2\}, i.e.,$ there exists $\delta > 0$ such that $g_u(x_1 | \{x_1, x_2\}; \psi) > g_u(x_2 | \{x_1, x_2\}; \psi) + \delta$
Predicting Choice with Set-Dependent Aggregation

<table>
<thead>
<tr>
<th>Set dependence</th>
<th>Extends (McFadden et al. (1973))</th>
<th>$w$</th>
<th>$r$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (IIA)</td>
<td>Tversky (1969)</td>
<td>(max-min)$^\nu$</td>
<td>zero</td>
<td>identity</td>
</tr>
<tr>
<td>Weights</td>
<td>McFadden (1978)</td>
<td>linear</td>
<td>log $\sum$ exp</td>
<td>log</td>
</tr>
<tr>
<td></td>
<td>Kalai et al. (2002)</td>
<td>softmax</td>
<td>zero</td>
<td>identity</td>
</tr>
<tr>
<td>Representations</td>
<td>Kaneko &amp; Nakamura (1979)</td>
<td>sum</td>
<td>min</td>
<td>log</td>
</tr>
<tr>
<td></td>
<td>Kivetz et al. (2004) (LAM)</td>
<td>sum</td>
<td>(max+min)/2</td>
<td>kinked lin.</td>
</tr>
<tr>
<td></td>
<td>Kivetz et al. (2004) (CCM)</td>
<td>sum</td>
<td>min</td>
<td>power($\rho$)</td>
</tr>
<tr>
<td>Both</td>
<td>Kivetz et al. (2004) (NCCM)</td>
<td>max-min</td>
<td>min</td>
<td>norm. pow($\rho$)</td>
</tr>
<tr>
<td></td>
<td>SDA (ours)</td>
<td>set-nn</td>
<td>set-nn</td>
<td>kinked tanh</td>
</tr>
</tbody>
</table>

Table 2: Discrete choice models as set-aggregation models.

2. otherwise $g$ is indifferent, i.e., for all other choice sets $s \subseteq \{x_1, x_2, x_3\}$ with $s \neq \{x_1, x_2\}$ and for all $x \in s$,
   
   $g_u(x|s; \psi) = c_s$

   for some constant $c_s$.

In terms of prediction, this means that $g_u$ predicts $x_1$ out of $\{x_1, x_2\}$, and is otherwise indifferent. Triple bases are useful in that claims regarding large collections of items can be established by reasoning about a single arbitrary triple of items (there are no restrictions on $x_1, x_2, x_3$).

The results of Ambrus & Rozen (2015) apply to aggregation mechanisms satisfying five properties that are standard in choice theory, and we will assume these hold for the mechanisms we consider as well. For some results, Ambrus & Rozen (2015) also require the mechanisms to be scale invariant (SI):

**Definition 4** (Scale invariance, Ambrus & Rozen (2015)).

An aggregation mechanism $\psi$ is **scale invariant** if there exists an odd and invertible function $\xi$ such that

$$\forall \alpha \in \mathbb{R}, \quad g_{\alpha f}(x|s; \psi) = \xi(\alpha)g_f(x|s; \psi)$$

where $\alpha f = \{\alpha f\}_{f \in F}$, i.e., functions in $F$ are scaled by $\alpha$.

Scale invariance states that the scale (or units) in which utility is stated does not change the predictive behavior of the aggregator. As conveyed in Ambrus & Rozen (2015), scale invariance is a useful property which holds for many known aggregators, and we focus on these here. The following result of Ambrus & Rozen (2015)—an excerpt from their main proof presented here as a lemma—is key to our proof:

**Lemma 1.** (Ambrus & Rozen, 2015) For all scale-invariant aggregation mechanisms $\psi$ there exist a triple basis $u$ with $\nu = 5$.

Note that our results also apply to some aggregators that are not scale invariant, albeit with a possibly larger $\nu$.

**B.2. Base class**

As noted in the main text, our proof requires that $G$ be defined over a base class of functions that is slightly more expressive than $F$. We denote this class $\mathcal{F}$ and define it here concretely. In general terms, each function $f \in \mathcal{F}$ can be thought of as composed of a pair of functions $f, f' \in F$ whose outputs are combined using simple operations. We will think of these operations as a small neural network $a(\cdot)$ with input of size 2 (taking in $f(x)$ and $f'(x)$) and having two hidden layers with two units each and with sigmoidal activations, and a final 2-to-1 linear layer (see Figure 5). We denote this class of auxiliary functions by $A$, and use it to define the base class:

$$\mathcal{F} = \{f(x; f, f', \alpha) = a(f(x), f'(x)) : f, f' \in F, a \in A\}$$

**B.3. Proof of Theorem 1**

We are now ready to give the proof for Theorem 1, revised and detailed below:

**Theorem 1** (Approximation error, revised). Let $k \geq 0$, and let $G$ be an aggregator class of dimension $\ell = 5k + 1$ over base class $\mathcal{F}$ and with a scale-invariant aggregation mechanism $\psi$ satisfying properties 1-5 of Ambrus & Rozen (2015). Then:

$$\min_{g \in \mathcal{G}} \varepsilon(g) \leq \min_{f_1, \ldots, f_\ell \in F} \sum_{i=0}^k p_i \min_{f_i \in \mathcal{F}} \varepsilon(f_i|C_i)$$

where $p_i = \mathbb{P}_D[s \in C_i]$, and $C_0, \ldots, C_k$ is the appropriate partition of $\mathcal{S}$ corresponding to $f_1', \ldots, f_\ell'$, i.e.,

$$C_i = \{s : \forall x \in s \ f_i'(x) > 0\} \setminus C_{i+1} \cup \cdots \cup C_k$$

for $i = 1, \ldots, k$, and $C_0 = \mathcal{S} \setminus C_1 \cup \cdots \cup C_k$.

At a high level, the proof consists of constructing an aggregator $g$ from two given collections of item score functions.
\[ f = (f_0, \ldots, f_k) \text{ and } f' = (f'_1, \ldots, f'_l) \] such that for all \( i = 0, \ldots, k, g \) will be as accurate as \( f_i \) on \( C_i \) (induced by \( f'_i \)). The optimal aggregator will then be at least as accurate as \( g \) for the optimal \( f, f' \). The core of the proof lies in designing small aggregation “modules” that implement a triple basis for various choice sets, and the final \( g \) is a linear combination of these modules with coefficients chosen to resolve any conflicts across modules.

**Proof.** Since \( \psi \) is scale invariant Lemma 1 states that there exists \( u = \{u_1, \ldots, u_3\}, u_i \in \mathbb{R}^3 \) that is a triple basis for it, with corresponding \( \delta \) (see Definition 3) and \( \xi \) (see Definition 4). As \( \psi \) is given (and is scale invariant), we fix throughout the proof \( u, \delta, \text{ and } \xi \). When clear from context we will drop the notational dependence of \( g \) on \( \psi \).

Our first step is to provide a sufficient condition under which \( u \) can be approximated by score functions.

**Definition 5.** Let \( \bar{f} = (\bar{f}_1, \ldots, \bar{f}_5), \bar{f}_i \in \mathcal{F}, \) then \( \bar{f} \) is an \( \epsilon \)-approximation of \( u \) if

\[
\max_{i,j} |\bar{f}_i(x_j) - u_{ij}| \leq \epsilon
\]

**Lemma 2.** Let \( f, f' \in \mathcal{F}, \) and let \( x_1, x_2, x_3 \in \mathcal{X} \) on which \( u \) is defined. If the following conditions hold:

1. \( f(x_1) > f(x_2) \)
2. \( f'(x_1), f'(x_2) > 0 \geq f'(x_3) \)

then \( u \) can be approximated by functions in \( \mathcal{F} \) to arbitrary precision, i.e., for all \( \epsilon > 0 \) exists \( \bar{f} \in \mathcal{F}^{(5)} \) that is an \( \epsilon \)-approximation of \( u \).

**Proof.** We first describe a general recipe for constructing a function \( \bar{f} \in \mathcal{F} \) from \( f, f' \in \mathcal{F} \) capable of approximating any vector \( u \in \mathbb{R}^3 \) when applied to (and with entries corresponding to) \( x_1, x_2, x_3 \), and then present the specific construction of \( \bar{f} \) for \( u \).

Let \( u \in \mathbb{R}^3 \) and fix \( \epsilon > 0 \). We now construct \( \bar{f} \in \mathcal{F} \) that approximates \( u \) on \( x_1, x_2, x_3 \). Functions in \( \mathcal{F} \) are of the form \( \bar{f}(x|f, f, \alpha) \), and so to make \( \bar{f} \) concrete, we must determine the parameters of \( \alpha \). Recall that each unit \( r \) takes in an input \( z \in \mathbb{R}^2 \), and applies an affine transformation:

\[ r(z; \alpha, \beta) = \langle \alpha, z \rangle + \beta, \ \alpha \in \mathbb{R}^2, \beta \in \mathbb{R} \]

We begin by determining the parameters \( \alpha, \beta \) of each of the hidden units \( r_i, i = 1, \ldots, 4 \), and then proceed to the final unit \( r_5 \) (see Figure 5). For hidden units, the affine transformation is followed by a sigmoidal activation, which we assume w.l.o.g. to be scaled to \([0, 1]\). We will use \( \approx \) to mean approximate to within an additive \( \epsilon \).

- The input of \( r_1 \) is \( z = (f(x), f'(x)) \). From condition 2, there exist \( \alpha, \beta \) such that \( \sigma(r_1(z; \alpha, \beta)) \approx 1 \) for \( x = x_1, x_2 \) and \( 0 \approx 0 \) for \( x = x_3 \). This is because \( \sigma \) is sigmoidal and hence \( \alpha \) and \( \beta \) can shift and scale the inputs to \( \sigma \) such that the higher-valued \( x_1, x_2 \) are “pushed” towards values arbitrarily close to 1, and the lower-valued \( x_3 \) towards values arbitrarily close to 0.

- The input of \( r_2 \) is also \( z = (f(x), f'(x)) \). From condition 1, there exist \( \alpha, \beta \) such that \( \sigma(r_2(z; \alpha, \beta)) \approx 1 \) for \( x = x_1 \) and \( 0 \approx 0 \) for \( x = x_2, x_3 \). This is because \( r_1 \) and \( r_2 \) contribute \( 1 \) to \( x_1 \), while \( x_2 \) and \( x_3 \) get at most one value that is near 1 from either \( r_1 \) or \( r_2 \).

- The input of \( r_3 \) is \( z = (r_1(x), r_2(x)) \). There exist \( \alpha, \beta \) such that \( \sigma(r_3(z; \alpha, \beta)) \approx 1 \) for \( x = x_1 \) and \( 0 \approx 0 \) for \( x = x_2, x_3 \). This is because \( r_1 \) and \( r_2 \) contribute \( 1 \) to \( x_1 \), while \( x_2 \) and \( x_3 \) get at most one value that is near 1 from either \( r_1 \) or \( r_2 \).

- The input of \( r_4 \) is also \( z = (r_1(x), r_2(x)) \). Because \( \sigma \) is sigmoidal, there exist \( \alpha \) with \( \alpha_2 = 0 \) such that with small enough \( \alpha_1 \) and with \( \beta = 0 \), \( r_4 \) can approximate the identity function on the first input, i.e., \( \sigma(r_4(z; \alpha, \beta)) \approx z_1 = r_1(x) \).

When \( \bar{f} \) is applied to \( x_1, x_2, x_3 \), the outputs of \( r_3 \) are approximately 1, 0, and 0, respectively, and the outputs of \( r_4 \) are approximately 1, 1, and 0, respectively. With slight abuse of notation we can think of the \( r_i \) as vector mappings (from \( \mathcal{X}^3 \) to \( \mathbb{R}^3 \)) and write:

\[
\begin{align*}
    r_3 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} & \approx \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, &
    r_4 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} & \approx \begin{pmatrix} 1 \\ 0 \end{pmatrix}
\end{align*}
\]

As \( r_3, r_4 \) are the inputs of \( r_5 \), we can also think of \( r_5 \) as a vector mapping whose bias term \( \beta \) acts on the unit vector:

\[
\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \approx \alpha_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \alpha_2 \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}
\]

\footnote{Alternatively, functions in \( \mathcal{A} \) can be defined with only one unit in the second layer, i.e., \( r_4 \) and its nonlinearity are removed, and instead \( r_5 \) takes as input the outputs of \( r_2 \) and \( r_3 \).}
Thus, with this construction of \( r_1, \ldots, r_4 \) and for the items \( x_1, x_2, x_3, r_5 \), we can think computing a linear combination of a linear basis of \( R^3 \). Hence, by setting its parameters, any vector \( u \in R^3 \) can be approximated.

Given the above, for each \( i = 1, \ldots, 5 \), we construct \( \bar{f}_i \) to approximate \( u_i \) by setting the parameters of its \( r_5 \) to \( \alpha_1 = u_{i1}, \alpha_2 = u_{i2}, \beta = u_{i3} \). Together, the resulting \( \bar{f} = (\bar{f}_1, \ldots, \bar{f}_5) \) approximates \( u \).

Since Lemma 3 allows for arbitrary approximations, we can choose \( \epsilon \) for which:

\[
\epsilon < \xi^{-1}(\delta/2^{k+2}) \quad (13)
\]

Since functions in \( \mathcal{F} \) can approximate triple bases, the properties of triple bases carry to aggregators.

**Corollary 1.** Let \( \bar{f} \) be as in Lemma 2, then:

1. \( g_{\bar{f}}(x_1 | \{ x_1, x_2 \}) > g_{\bar{f}}(x_2 | \{ x_1, x_2 \}) + \delta (1 - \frac{1}{2^k}) \)

2. On all other sets \( s \subseteq \{ x_1, x_2, x_3 \} \), \( s \neq \{ x_1, x_2 \} \) there is approximate indifference, i.e., for all such \( s \) and for all \( x, x' \in s \),

\[
|g_{\bar{f}}(x|s) - g_{\bar{f}}(x'|s)| < \frac{\delta}{2^{k+1}}
\]

The above follows from the definition of triple bases (Lemma 1), from scale invariance, and from Eq. (13). Note that scale invariance ensures that an error of at \( \epsilon \) in the approximation of \( u \) results in a “propagated” error of at most \( \xi(\epsilon) \) when passed through an aggregator.

Next, we move away from choice sets of size three, and consider how functional approximations of triple bases operate on general choice sets.

**Definition 6.** Let \( f' \in \mathcal{F} \). The collection of choice sets that are separated by \( f' \) is:

\[
\Omega_{f'} = \{ s \in \mathcal{S} : f'(s) > 0 \}
\]

As before, \( f'(s) > 0 \) holds if \( f'(x) > 0 \) for all \( x \in s \).

**Lemma 3.** Let \( f, f' \in \mathcal{F} \) and let \( \bar{f} \in \mathcal{F}^{(5)} \) be as in Lemma 2. Let \( s \in \mathcal{S} \), and denote \( x^* = \arg \max_{x \in s} f(x) \). Then if \( s \in \Omega_{f'} \), it holds that:

\[
\forall x \in s, x \neq x^*, \quad g_{\bar{f}}(x^*|s) > g_{\bar{f}}(x|s) + \delta (1 - \frac{1}{2^k})
\]

Otherwise, if \( s \notin \Omega_{f'} \), then:

\[
\forall x, x' \in s, \quad |g_{\bar{f}}(x|s) - g_{\bar{f}}(x'|s)| < \frac{\delta}{2^{k+1}}
\]

**Proof.** Consider first the triple basis \( u \). As noted in Ambruś & Rozen (2015), triple bases can be applied to a set \( s \) by associating (i.e., assigning the utility of) the predicted item \( \hat{y} \in s \) with \( x_1 \), associating the other alternatives \( x \in s \) with \( x_2 \), and associating all other items \( X \setminus s \) with \( x_3 \). In this way, \( g_u \) will predict \( \hat{y} \) out of any \( s' \subset s \) for which \( \hat{y} \in s' \), and be indifferent otherwise.

When aggregation is applied to score functions rather than utility vectors via \( g_f \), associations are determined by \( f \) and \( f' \). \( f' \) determines which items are associated with \( x_1 \) or \( x_2 \) and which with \( x_3 \), and \( f \) determines the item associated with \( x_1 \). By construction, \( f' \) associates with \( x_3 \) exactly those items \( x \in X \) for which \( f'(x) \leq 0 \), and so \( g_f \) is indifferent to all \( s \notin \Omega_f \). Meanwhile, for all \( s \in \Omega_f \), \( f'(s) > 0 \) and so \( g_{\bar{f}} \) is not indifferent, and predictions are determined by \( f \) through its association of items with \( x_1 \).

The conclusion from Lemma 3 is that, by approximating a triple basis on sets, \( \bar{f} \) agrees with \( f \) on all sets separated by \( f' \), and is (approximately) indifferent otherwise.

We are now ready to construct the final aggregator \( g \). The aggregator will be composed of a collection of “modules”—small aggregators \( g^{(i)} \), \( i = 0, \ldots, k \), each targeting a different element of the partition.

Let \( f_0, \ldots, f_k \in \mathcal{F} \), \( f_1', \ldots, f_k' \in \mathcal{F} \). For each \( i = 1, \ldots, k \), consider \( \bar{f}_i \in \mathcal{F}^{(5)} \) constructed from \( f_i, f_i' \) as in Lemma 2. For \( i = 0 \), let \( \bar{f}_0 \) include a single function \( f_0 \in \mathcal{F} \), for which \( f_0(x) = f_0(x) \).

The modules are defined by:

\[
g^{(i)} = g_{\bar{f}_i} \quad (14)
\]

and the aggregator by:

\[
g(x|s) = \sum_{i=0}^{k} \alpha_i g^{(i)}(x|s) \quad (15)
\]

where coefficients are set by:

\[
\alpha_i = 1/2^{k-i+1}, \quad i = 0, \ldots, k \quad (16)
\]

Note that \( g \) is indeed an aggregator due to the following closure properties of aggregators:

- Since \( \psi \) is scale-invariant, and since \( \mathcal{F} \) is closed under scalar multiplication, \( \mathcal{G} \) is also closed under scalar multiplication, i.e., if \( g \in \mathcal{G} \), then also \( \alpha g \in \mathcal{G} \) for any \( \alpha \in R \).

---

8This can be achieved by setting the parameters of \( r_2, r_4 \) to 0, setting \( r_1, r_3 \) to approximate the identity function (as \( r_4 \) in Lemma 2), and \( r_5 \) to match \((0, 1, 0)\).
we will consider the contribution of each module \( g \), which

The following lemma establishes how \( g \) operates on each
region \( C_i \) of the partition \( C_0, \ldots, C_k \). In particular, it shows how the coefficients \( \alpha_i \) determine an order of precedence over the modules \( g^{(i)} \) that prevents collisions: if a choice set \( s \) is separated by more than one module, then the coefficients ensure that it will be taken into account only in the \( C_i \) for which \( \alpha_i \) is largest.

**Lemma 4.** The aggregator \( g \) from Eq. (15) agrees on each
\( C_i \) with the corresponding \( g^{(i)} \), i.e., if \( s \in C_i \), then

\[
\arg\max_{x \in s} g(x|s) = \arg\max_{x \in s} g^{(i)}(x|s)
\]

**Proof.** Consider some \( s \in S \). Let \( i \) be the maximal index
such that \( s \in \Omega_{f_i} \), and note that this implies \( s \in C_i \). Denote
\( x^* = \arg\max_{x \in s} g^{(i)}(s) = \arg\max_{x \in s} f_i(s) \). We would
like to show that also \( x^* = \arg\max_{x \in s} g(x|s) \). To do this, we
will consider the contribution of each module \( g^{(j)} \) to
each item \( x \in s \), and show that the cumulative contribution to \( x^* \) (and hence its relative value under \( g \)) is larger than that of any other item.

For module \( i \), because \( s \in \Omega_{f_i} \), from Lemma 3 we have that
\( g^{(i)} \) scores \( x^* \) higher than any other \( x \in s \) by a margin of
at least \( \delta(1 - 1/2^{k+1}) = \delta(1 - \alpha_0) \). Hence, the weighted
contribution of module \( i \) to \( x^* \) is higher than its contribution
to all other items by at least:

\[
\delta \alpha_i (1 - \alpha_0)
\]

Consider the maximal contribution of other modules to some
\( x \neq x^* \). For any module \( j < i \), from Lemma 3 we have that
the contribution of \( g^{(j)} \) to \( x \) is either \( \delta(1 - \alpha_0) \) (if \( s \in \Omega_{f_j} \))
or \( \delta \alpha_0 \) (if \( s \notin \Omega_{f_j} \)), and so in any case is at most \( \delta \). Hence,
the combined weighted contributions of all \( j < i \) is at most:

\[
\sum_{j < i} \alpha_j g^{(j)}(x|s) = \delta \sum_{j < i} \alpha_j \leq \delta \alpha_i - \alpha_0
\]

Meanwhile, for any module \( j > i \), since \( i \) is maximal, from Lemma 3 we have that the contribution of \( g^{(j)} \) to \( x \) is at most \( \delta/2^{k+1} = \delta \alpha_0 \). Hence, the combined weighted
contributions of all \( j > i \) is at most:

\[
\sum_{j > i} \alpha_j g^{(j)}(x|s) \leq \delta \alpha_0 \sum_{j > i} \alpha_j \leq \delta \alpha_0 (1 - 2 \alpha_i)
\]

Overall, the combined weighted contributions of all modules
\( j \neq i \) to any \( x \neq x^* \) sum to at most

\[
\delta \alpha_i (1 - 2 \alpha_0)
\]

which is strictly less then the contribution of \( g^{(i)} \) to \( x^* \) (Eq.
(17)), and so \( \arg\max_{x \in s} g(x|s) = x^* \).

For the last step, let each \( f_i \) be locally optimal for \( C_i \), i.e.,
\( f_i = \arg\min_{f \in F} \varepsilon(f|C_i) \). Hence, for \( g^{(i)} \) defined w.r.t.
this \( f_i \),

\[
\varepsilon(g^{(i)}|C_i) \leq \min_{f \in F} \varepsilon(f|C_i)
\]

Since \( C_0, \ldots, C_k \) form a partition, from Lemma 4 and from
the optimality of \( g^* \) we have that:

\[
\varepsilon(g^*) \leq \varepsilon(g) \leq \sum_{i=1}^k p_i \min_{f \in F} \varepsilon(f|C_i) \tag{19}
\]

Finally, considering the optimal \( f'_1, \ldots, f'_k \) for the partition
\( C_0, \ldots, C_k \) entails that Eq. (19) holds for all appropriate
partitions, thus concluding the proof.

**C. Estimation Error**

We begin with Theorem 2 which considers set-dependent
weight aggregators of the form \( g(x|s) = \langle w(s), \phi(x) \rangle \) where:

\[
\phi(x) = (\tilde{x}_1, \ldots, \tilde{x}_k)
\]

\[
w(s) = (w_1(s), \ldots, w_k(s))
\]

where \( w_i(s) = w(\tilde{s}_i) \) and \( \tilde{x}_i = f_i(x) = (\theta_i, x) \).

We can write \( \phi(x) = x^\top \Theta \) where \( \Theta_i, = \theta_i \) are rows, and
denote columns by \( \theta_j = \Theta_j \). Note that:

\[
g(x|s) = \sum_{i=1}^d w_i(s) \langle \theta_i, x \rangle = \sum_{k=1}^d \sum_{i=1}^\ell \Theta_{ik} w_i(s) x_k
\]

\[
= \sum_{k=1}^d (\bar{\theta}_k, w(s)|x_k)_{(d)} \tag{III}
\]

We now bound the Rademacher complexity of each component.
Most inequalities follow from the decomposition rules in
Shalev-Shwartz & Ben-David (2014) (specific lemmas
therein referenced in brackets).

\[
R_f \leq X \alpha \sqrt{\frac{2 \log 2d}{m}} \tag{Lemma 26.11}
\]

\[
R_w \leq \lambda^p R_f \tag{Lemma 26.9}
\]

\[
R_{(d)} \leq \max_x \| x \|_{\infty} R_w \tag{Lemma 26.6}
\]

\[
R_{(d)} \leq 2 \max_{\theta} \| \bar{\theta} \|_1 \cdot R_{(d)} = 2 \| \Theta \|_1 R_{(d)}
\]
where the last inequality follows from Sec. 4 in Sridharan (2014). Assuming \( \|\Theta\|_1 \leq 1 \), combining the above gives:

\[
R_g \leq 2X_\infty^2 \lambda_\epsilon^g \sqrt{\frac{2\log 2d}{m}}
\]

Using standard Rademacher-based generalization bounds (e.g., Bartlett & Mendelson (2002)) concludes the proof:

\[
\varepsilon(G) \leq \varepsilon_T(G) + 2R_g + O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right)
\]  \hspace{1cm} (20)

We next turn to Theorem 3 which considers set-dependent embedding aggregators of the form \( g(x|s) = \langle v, \phi(x|s) \rangle \) where \( v \in \mathbb{R}^d \) and \( \phi(x|s) = \mu(x - r(s)) \) where:

\[
r(s) = (r_1(s), \ldots, r_d(s)), \quad r_i(s) = r(\tilde{s}_i)
\]

Inequalities again follow Shalev-Shwartz & Ben-David (2014). The Rademacher complexity of each component is:

\[
R_\mu \leq \lambda_\mu(R_f + R_\varepsilon) \quad \text{(Lemma 26.6)}
\]

\[
R_\varepsilon \leq \lambda_\varepsilon^\rho \quad \text{(Lemma 26.6)}
\]

\[
R_g \leq 2W_1 R_\mu \quad \text{(Lemma 26.7)}
\]

and \( R_f \) is as before. Together, this gives:

\[
R_g \leq 2W_1 R_\mu \leq 2W_1 \lambda_\mu(1 + \lambda_\epsilon^g)X_\infty \sqrt{\frac{2\log 2d}{m}}
\]

Applying the generalization bound in Eq. (20) concludes our proof.

D. Experiments

D.1. Datasets

1. Amadeus: Each item is a flight itinerary, and choice sets include a collection of recommended itineraries. Choice corresponds to clicking on an item, and examples include choice sets having exactly one click. Features include for example flight origin/destination, price, and number of transfers (for a full list see Mottini & Acuna-Agost (2017)). User features are excluded from the original dataset due to privacy concerns.

2. Expedia: Each item is a recommended hotel, and choice sets include items corresponding to a search result. We use the non-randomized portion of the dataset (see original data description for details). Choice corresponds to clicking on an item, and examples include choice sets having exactly one click. Features include for example hotel price, rating, length of stay, booking window, user’s average past ratings, and visitor’s location. We applied the following standard preprocessing steps for different variable types:

\begin{itemize}
  \item Continuous: for some features, a log or square-root transform.
  \item Ordinal: One-hot encoding.
  \item Date/time: use week and month to capture seasonality of hotel pricing.
  \item Categorical: One-hot encoding. For features with a large number of categories, top categories where encoded directly, while the rest were binned into a single variable.
  \item Additional features: popularity score.\(^9\)
\end{itemize}

3. Outbrain: Each item is a news article, and choice sets include a collection of recommended items. Choice corresponds to clicking on an item, and examples include choice sets having exactly one click. Recommendations are given to user within the context of a currently viewed news article, and so features describe both current and recommended articles. Features include for example article category, advertiser ID, and geo-location of the views. We applied several preprocessing steps.\(^10\).

Table 3 includes summary statistics for each dataset:

| Dataset     | m    | |X| | max(n) | avg(n) | d |
|-------------|------|-----|------|--------|-------|---|
| Amadeus     | 34K  | 1.0M| 50   | 32.1   | 17    |
| Expedia     | 199K | 129K| 38   | 25     | 8     |
| Outbrain    | 16.8M| 478K| 12   | 5.2    | 10    |

Code for preprocessing the Expedia and Outbrain datasets as in our experiments can be found in our online code repository. For the Amadeus data, please see Mottini & Acuna-Agost (2017).

D.2. Baselines

\begin{itemize}
  \item MNL: Our implementation.
  \item SVMRank: Open-source code with minor modifications.\(^11\)
  \item RankNet: Learning2rank library.\(^12\)
  \item MixedMNL: Our implementation.
  \item AdaRank: Open-source code.\(^13\)
\end{itemize}


\(^10\)We followed steps 1-5 https://github.com/alexeygrigorev/outbrain-click-prediction-kaggle

\(^11\)https://gist.github.com/coreylynch/4150976/

\(^12\)https://github.com/shiba24/learning2rank

\(^13\)https://github.com/rueycheng/AdaRank
Predicting Choice with Set-Dependent Aggregation

- **DeepSets**: Source code provided by the authors.\(^{14}\)

**Number of parameters.** For neural network based models SDA, RankNet, Deep Sets, the number of parameters are \(816 + 784d, 525312 + 1024d, 196864 + 256d\), respectively, where \(d\) is the number of features in a dataset. For a reasonable range of \(d\), the number of SDA parameters is significantly lower than that of other models. This further illustrates how SDA reduces model complexity by incorporating inductive bias in clever ways. Table 4 includes the number of parameters per dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SDA</th>
<th>RankNet</th>
<th>Deep Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amadeus</td>
<td>14,144</td>
<td>542,720</td>
<td>201,216</td>
</tr>
<tr>
<td>Expedia</td>
<td>7,088</td>
<td>533,504</td>
<td>198,912</td>
</tr>
<tr>
<td>Outbrain</td>
<td>8,656</td>
<td>535,552</td>
<td>199,424</td>
</tr>
</tbody>
</table>

**D.3. Experimental setup**

**Implementation.** SDA was implemented in Python and using Tensorflow\(^{15}\). Our code is open source and publicly available, please refer to to the author’s website for an updated link to the repository.

**Hyperparameters.** For all methods, we tuned regularization, dropout, and learning rate (when applicable) using Bayesian optimization using the open source library Optuna\(^{16}\). Hyperparameters were tuned on a held-out validation set for 100 trials of Bayesian optimization. Sampling ranges include:

- **Learning rate**: \([10^{-5}, 10^{-3}]\) (log-uniform sampling)
- **Weight decay**: \([10^{-10}, 10^{-3}]\) (log-uniform sampling)
- **Dropout rate**: \([0.5, 1]\) (uniform sampling)

We used exponential decay with decay rate of 0.95 with decay step of 10 for all models. All models were trained with a batch size of 128, and early stopping was done based on the validation accuracy with an early stop window of 25 epochs. All tuning and training was done on CPUs.

**D.4. SDA**

Most of the literature on choice models considers scalar score functions, scalar reference points, and comparison via negation. We experiment with a broader notion of comparison that differs in two ways. First, values and references are multi-dimensional. Here, we use vector-valued score functions \(F : A^k \rightarrow \mathbb{R}^k\) and vector-valued reference functions \(r : 2^k \rightarrow \mathbb{R}^k\) for some \(k\). Second, value-reference comparisons within each dimension of aggregation \(i \in [\ell]\) are done using an inner product \(\langle \hat{x}, r(\hat{s}) \rangle\), with multi-dimensional embedded “dimensions” \(\hat{x}_i \in \mathbb{R}^k\) and reference points \(r(\hat{s}_i) \in \mathbb{R}^k\), \(\hat{s}_i = \{\hat{x}_i\}_{i \in A}\). We denote this model by SDA and use it in the following sections.

**D.5. Additional choice set sizes**

Results in the main paper correspond to item sets with at most 10 items for Amadeus and Expedia and 12 for Outbrain. We further experimented with choice sets of larger maximum sizes:

1. **Amadeus**: 10, 20, 30, 40, 50 (max)
2. **Expedia**: 10, 20, 30, 38 (max)
3. **Outbrain**: 12 (max)

Result are given in Table 6. We report top-1 accuracy, top-5 accuracy, and mean rank.

**D.6. Ablation Study**

We performed an extensive ablation study, demonstrating the contribution of each component of SDA in an ablation study and justifying our modeling decisions. Table 5 includes the configurations of all ablated models. Of particular interest are models that, in line with Sec. 2.2, have only set dependent weights (SDW) or representations (SDR):

**Table 5:** Specification of all ablated models

<table>
<thead>
<tr>
<th>(\ell)</th>
<th>(w)</th>
<th>(\phi)</th>
<th>(r)</th>
<th>(\mu)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDA(_{\perp})</td>
<td>24</td>
<td>Set NN</td>
<td>(\langle \hat{x}, r(\hat{s}) \rangle)</td>
<td>Set NN (\circ)-tanh</td>
</tr>
<tr>
<td>SDA(_{\perp}), (\mu = \tanh)</td>
<td>24</td>
<td>Set NN</td>
<td>(\langle \hat{x}, r(\hat{s}) \rangle)</td>
<td>Set NN \tanh</td>
</tr>
<tr>
<td>SDA(_{\perp}), no (\mu)</td>
<td>24</td>
<td>Set NN</td>
<td>(\langle \hat{x}, r(\hat{s}) \rangle)</td>
<td>Set NN -</td>
</tr>
<tr>
<td>SDR</td>
<td>24</td>
<td>(\in \mathbb{R}^\ell)</td>
<td>(\langle \hat{x}, r(\hat{s}) \rangle)</td>
<td>Set NN (\circ)-tanh</td>
</tr>
<tr>
<td>SDW</td>
<td>24</td>
<td>Set NN</td>
<td>(\hat{x})</td>
<td>-</td>
</tr>
<tr>
<td>SDW, single (f)</td>
<td>1</td>
<td>Set NN</td>
<td>(\hat{x})</td>
<td>-</td>
</tr>
<tr>
<td>MNL</td>
<td>1</td>
<td>(= 1)</td>
<td>(\hat{x})</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 7 shows results for all ablated models and on all choice set sizes. The experimental setup follows that of Sec. 4.
Table 6: Full experimental results.

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1</th>
<th>Top-5</th>
<th>Mean rank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDAa</td>
<td>±0.14</td>
<td>±0.13</td>
<td>±0.14</td>
</tr>
<tr>
<td>MNL</td>
<td>±0.14</td>
<td>±0.13</td>
<td>±0.14</td>
</tr>
<tr>
<td>SVMRank</td>
<td>±0.14</td>
<td>±0.13</td>
<td>±0.14</td>
</tr>
<tr>
<td>RankNet</td>
<td>±0.14</td>
<td>±0.13</td>
<td>±0.14</td>
</tr>
<tr>
<td>Mixed MNL</td>
<td>±0.14</td>
<td>±0.13</td>
<td>±0.14</td>
</tr>
<tr>
<td>AdaRank</td>
<td>±0.14</td>
<td>±0.13</td>
<td>±0.14</td>
</tr>
<tr>
<td>Deep Sets</td>
<td>±0.14</td>
<td>±0.13</td>
<td>±0.14</td>
</tr>
<tr>
<td>Price/Quality</td>
<td>±0.14</td>
<td>±0.13</td>
<td>±0.14</td>
</tr>
<tr>
<td>Random</td>
<td>±0.14</td>
<td>±0.13</td>
<td>±0.14</td>
</tr>
</tbody>
</table>

Top-1: 10, 20, 30, 40, 50
Top-5: 10, 20, 30, 40, 50
Mean rank: 10, 20, 30, 40, 50
Table 7: Ablation Experiment Result.

<table>
<thead>
<tr>
<th>Amadeus</th>
<th>Top-1</th>
<th>Top-5</th>
<th>Mean rank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>SDA,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>45.42</td>
<td>33.48</td>
<td>29.26</td>
</tr>
<tr>
<td>SDA, w/ ( \mu ) = tanh</td>
<td>39.10</td>
<td>31.54</td>
<td>27.02</td>
</tr>
<tr>
<td>SDA, w/ no ( \mu )</td>
<td>41.38</td>
<td>33.25</td>
<td>28.89</td>
</tr>
<tr>
<td>SDR</td>
<td>37.50</td>
<td>30.64</td>
<td>26.16</td>
</tr>
<tr>
<td>SDW</td>
<td>39.98</td>
<td>32.03</td>
<td>27.85</td>
</tr>
<tr>
<td>SDW w/ single ( f )</td>
<td>38.21</td>
<td>27.60</td>
<td>23.66</td>
</tr>
<tr>
<td>MNL</td>
<td>38.42</td>
<td>27.93</td>
<td>23.54</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expedia</th>
<th>Top-1</th>
<th>Top-5</th>
<th>Mean rank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>SDA,</td>
<td>31.49</td>
<td>26.81</td>
<td>21.96</td>
</tr>
<tr>
<td>SDA, w/ ( \mu ) = tanh</td>
<td>31.19</td>
<td>26.25</td>
<td>21.81</td>
</tr>
<tr>
<td>SDA, w/ no ( \mu )</td>
<td>30.90</td>
<td>26.52</td>
<td>21.88</td>
</tr>
<tr>
<td>SDR</td>
<td>25.05</td>
<td>22.43</td>
<td>18.24</td>
</tr>
<tr>
<td>SDW</td>
<td>31.27</td>
<td>26.80</td>
<td>21.88</td>
</tr>
<tr>
<td>SDW w/ single ( f )</td>
<td>30.04</td>
<td>25.28</td>
<td>20.82</td>
</tr>
<tr>
<td>MNL</td>
<td>30.06</td>
<td>25.29</td>
<td>20.61</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outbrain</th>
<th>Top-1</th>
<th>Top-5</th>
<th>Mean rank</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>SDA,</td>
<td>86.91</td>
<td>73.08</td>
<td>61.68</td>
</tr>
<tr>
<td>SDA, w/ ( \mu ) = tanh</td>
<td>86.18</td>
<td>72.61</td>
<td>60.90</td>
</tr>
<tr>
<td>SDA, w/ no ( \mu )</td>
<td>86.31</td>
<td>72.80</td>
<td>60.79</td>
</tr>
<tr>
<td>SDR</td>
<td>80.67</td>
<td>68.61</td>
<td>55.40</td>
</tr>
<tr>
<td>SDW</td>
<td>86.67</td>
<td>73.08</td>
<td>61.07</td>
</tr>
<tr>
<td>SDW w/ single ( f )</td>
<td>85.41</td>
<td>71.88</td>
<td>58.98</td>
</tr>
<tr>
<td>MNL</td>
<td>86.34</td>
<td>72.90</td>
<td>60.94</td>
</tr>
</tbody>
</table>