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 μ 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_i 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 $r(\tilde{s}_1)$ (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 $\tilde{x}_i - r(\tilde{s}_i)$.

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 μ. 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 that we will 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 $\boldsymbol{u} = \{u_1, \dots, u_\nu\}$ where each $u_i \in \mathbb{R}^N$, we slightly abuse notation and use $g_{\boldsymbol{u}}$ to denote an aggregator with item-wise score functions corresponding to the utilities in $\boldsymbol{u}$.

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 $x_1, x_2, x_3 \in \mathcal{X}$. Let $\nu \in \mathbb{N}$, and let $\boldsymbol{u} = \{u_1, \dots, u_\nu\}$,$u_i \in \mathbb{R}^3$. Then $\boldsymbol{u}$ is a* **triple basis** *for x_1, x_2, x_3 under the aggregation mechanism ψ 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*

$$g_{\boldsymbol{u}}(x_1|\{x_1, x_2\}; \psi) > g_{\boldsymbol{u}}(x_2|\{x_1, x_2\}; \psi) + \delta$$

Set dependence	Extends	w	r	μ
None (IIA)	MNL (McFadden et al. (1973))	one	zero	identity
Weights	Tversky (1969)	(max-min)$^\rho$	zero	identity
	McFadden (1978)	linear	$\log \sum \exp$	log
	Kalai et al. (2002)	softmax	zero	identity
	Orhun (2009)	linear	w. average	kinked lin.
Representations	Kaneko & Nakamura (1979)	sum	min	log
	Kivetz et al. (2004) (LAM)	sum	(max+min)/2	kinked lin.
	Kivetz et al. (2004) (CCM)	sum	min	power(ρ)
Both	Kivetz et al. (2004) (NCCM)	max-min	min	norm. pow(ρ)
	SDA (ours)	set-nn	set-nn	kinked tanh

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 $x \in s$,*

$$g_{\boldsymbol{u}}(x|s;\psi) = c_s$$

for some constant c_s.

In terms of prediction, this means that $g_{\boldsymbol{u}}$ 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 ψ is* **scale invariant** *if there exists an odd and invertible function ξ such that*

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

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

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 ψ there exist a triple basis $\boldsymbol{u}$ with $\nu = 5$.*

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

B.2. Base class

As noted in the main text, our proof requires that $\mathcal{G}$ be defined over a base class of functions that is slightly more expressive than $\mathcal{F}$. We denote this class $\bar{\mathcal{F}}$ and define it here concretely. In general terms, each function $\bar{f} \in \bar{\mathcal{F}}$ can be thought of as composed of a pair of functions $f, f' \in \mathcal{F}$ 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 $\mathcal{A}$, and use it to define the base class:

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

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 $\mathcal{G}$ be an aggregator class of dimension $\ell = 5k+1$ over base class $\bar{\mathcal{F}}$ and with a scale-invariant aggregation mechanism ψ satisfying properties 1-5 of Ambrus & Rozen (2015). Then:*

$$\min_{g \in \mathcal{G}} \varepsilon(g) \leq \min_{f'_1, \dots, f'_k \in \mathcal{F}} \sum_{i=0}^{k} p_i \min_{f_i \in \mathcal{F}} \varepsilon(f_i | C_i)$$

where $p_i = \mathbb{P}_D[s \in C_i]$, and $C_0, \dots, C_k$ is the appropriate partition of $\mathcal{S}$ corresponding to $f'_1, \dots, f'_k$, i.e.,

$$C_i = \{s \,:\, \forall x \in s \;\, f'_i(x) > 0\} \setminus C_{i+1} \cup \dots \cup C_k$$

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

At a high level, the proof consists of constructing an aggregator g from two given collections of item score functions

$\boldsymbol{f} = (f_0, \dots, f_\ell)$ and $\boldsymbol{f}' = (f'_1, \dots, f'_\ell)$ such that for all $i = 0, \dots, k$, 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 $\boldsymbol{f}, \boldsymbol{f}'$. 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 ψ is scale invariant Lemma 1 states that there exists $\boldsymbol{u} = \{u_1, \dots, u_5\}$, $u_i \in \mathbb{R}^3$ that is a triple basis for it, with corresponding δ (see Definition 3) and ξ (see Definition 4). As ψ is given (and is scale invariant), we fix throughout the proof $\boldsymbol{u}$, δ, and ξ. When clear from context we will drop the notational dependence of g on ψ.

Our first step is to provide a sufficient condition under which $\boldsymbol{u}$ can be approximated by score functions.

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

$$\max_{i,j} |\bar{f}_i(x_j) - u_{ij}| \le \epsilon$$

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

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

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

Proof. We first describe a general recipe for constructing a function $\bar{f} \in \bar{\mathcal{F}}$ 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 $\bar{\boldsymbol{f}}$ for $\boldsymbol{u}$.

Let $u \in \mathbb{R}^3$ and fix $\epsilon > 0$. We now construct $\bar{f} \in \bar{\mathcal{F}}$ that approximates u on x_1, x_2, x_3. Functions in $\bar{\mathcal{F}}$ are of the form $\bar{f}(x|f, f, a)$, and so to make $\bar{f}$ concrete, we must

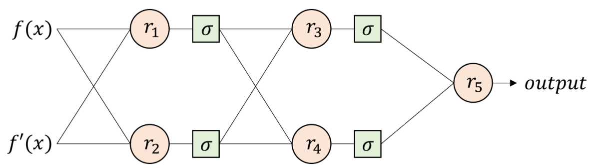

Figure 5: A function $a(\cdot)$ from the auxiliary class $\mathcal{A}$. Each unit r_i is an affine transformation $r_i(z) = \langle \alpha_i, z \rangle = \beta_i$ with parameters $\alpha_i \in \mathbb{R}^2, \beta_i \in \mathbb{R}$, and σ is a sigmoidal activation.

determine the parameters of a. Recall that each unit r takes in an input $z \in \mathbb{R}^2$, and applies an affine transformation:

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

We begin by determining the parameters α, β of each of the hidden units $r_i, i = 1, \dots, 4$, 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 ϵ.

- The input of r_1 is $z = (f(x), f'(x))$. From condition 2, there exist α, β such that $\sigma(r_1(z; \alpha, \beta)) \approx 1$ for $x = x_1, x_2$ and ≈ 0 for $x = x_3$. This is because σ is sigmoidal and hence α and β can shift and scale the inputs to σ 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 α, β such that $\sigma(r_2(z; \alpha, \beta)) \approx 1$ for $x = x_1$ and to ≈ 0 for $x = x_2$. Note that this gives no guarantees as to the output for $x = x_3$, but due to σ it is in $[0, 1]$.
- The input of r_3 is $z = (r_1(x), r_2(x))$. There exist α, β such that $\sigma(r_3(z); \alpha, \beta)) \approx 1$ for $x = x_1$ and ≈ 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 σ is sigmoidal, there exist α with $\alpha_2 = 0$ such that with small enough α_1 and with $\beta = 0$, r_4 can approximate the identity function on the first input, i.e., $\sigma(r_4(z; \alpha, \beta)) \approx z_1 = r_1(x)$.[7]

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:

$$r_3 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \approx \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad r_4 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \approx \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \tag{12}$$

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 β acts on the unit vector:

$$r_5 \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \approx \alpha_1 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \alpha_2 \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

[7] 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 $r_1, \dots, r_4$ and for the items x_1, x_2, x_3, r_5 can be thought computing a linear combination of a linear basis of R^3. Hence, by setting its parameters, any vector $u \in \mathbb{R}^3$ can be approximated.

Given the above, for each $i = 1, \dots, 5$, 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 $\bar{\boldsymbol{f}} = (\bar{f}_1, \dots, \bar{f}_5)$ approximates $\boldsymbol{u}$. ☐

Since Lemma 3 allows for arbitrary approximations, we can choose ϵ for which:

$$\epsilon < \xi^{-1}(\delta/2^{k+2}) \tag{13}$$

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

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

1. $g_{\bar{\boldsymbol{f}}}(x_1|\{x_1, x_2\}) > g_{\bar{\boldsymbol{f}}}(x_2|\{x_1, x_2\}) + \delta(1 - \frac{1}{2^{k+1}})$
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 $x, x' \in s$,*

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

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 ϵ in the approximation of $\boldsymbol{u}$ 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 $\bar{\boldsymbol{f}} \in \bar{\mathcal{F}}^{(5)}$ be as in Lemma 2. Let $s \in \mathcal{S}$, and denote $x^* = \operatorname{argmax}_{x \in s} f(x)$. Then if $s \in \Omega_{f'}$, it holds that:*

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

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

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

Proof. Consider first the triple basis $\boldsymbol{u}$. As noted in Ambrus & 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 $\mathcal{X} \setminus s$ with x_3. In this way, $g_{\boldsymbol{u}}$ 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 vetors via $g_{\bar{\boldsymbol{f}}}$, 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 $x \in \mathcal{X}$ for which $f'(x) \leq 0$, and so $g_{\bar{\boldsymbol{f}}}$ is indifferent to all $s \notin \Omega_{f'}$. Meanwhile, for all $s \in \Omega_{f'}$, $f'(s) > 0$ and so $g_{\bar{\boldsymbol{f}}}$ 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, $\bar{\boldsymbol{f}}$ 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 $g^{(i)}$, $i = 0, \dots, k$, each targeting a different element of the partition.

Let $f_0, \dots, f_\ell \in \mathcal{F}$, $f'_1, \dots, f'_\ell \in \mathcal{F}$. For each $i = 1, \dots, k$, consider $\bar{\boldsymbol{f}}_i \in \bar{\mathcal{F}}^{(5)}$ constructed from f_i, f'_i as in Lemma 2. For $i = 0$, let $\bar{\boldsymbol{f}}_0$ include a single function $\bar{f}_0 \in \bar{\mathcal{F}}$ for which $\bar{f}_0(x) = f_0(x)$.[8]

The modules are defined by:

$$g^{(i)} = g_{\bar{\boldsymbol{f}}_i} \tag{14}$$

and the aggregator by:

$$g(x|s) = \sum_{i=0}^{k} \alpha_i g^{(i)}(x|s) \tag{15}$$

where coefficients are set by:

$$\alpha_i = 1/2^{k-i+1}, \qquad i = 0, \dots, k \tag{16}$$

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

- Since ψ is scale-invariant, and since $\bar{\mathcal{F}}$ 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 $\alpha \in \mathbb{R}$.

[8] This 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 $(0, 1, 0)$.

- Aggregator classes are closed under addition in the following sense: denote by $g \in \mathcal{G}^{(n)}$ a class of aggregators of dimension n, then if $g \in \mathcal{G}^{(n)}$ and $g' \in \mathcal{G}^{(n')}$, it holds that $g + g' \in \mathcal{G}^{(n+n')}$.

The following lemma establishes how g operates on each region C_i of the partition $C_0, \dots, C_k$. In particular, it shows how the coefficients α_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 α_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*

$$\operatorname*{argmax}_{x \in s} g(x|s) = \operatorname*{argmax}_{x \in s} g^{(i)}(x|s)$$

Proof. Consider some $s \in \mathcal{S}$. Let i be the maximal index such that $s \in \Omega_{f'_i}$, and note that this implies $s \in C_i$. Denote $x^* = \operatorname{argmax}_{x \in s} g^{(i)}(s) = \operatorname{argmax}_{x \in s} f_i(s)$. We would like to show that also $x^* = \operatorname{argmax}_{x \in s} g(x|s)$. 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:

$$\delta \alpha_i (1 - \alpha_0) \tag{17}$$

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 δ. Hence, the combined weighted contributions of all $j < i$ is at most:

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

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:

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

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

$$\delta \alpha_i (1 - 2\alpha_0) \tag{18}$$

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

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

$$\varepsilon(g^{(i)}|C_i) \le \min_{f \in \mathcal{F}} \varepsilon(f|C_i)$$

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

$$\varepsilon(g^*) \le \varepsilon(g) \le \sum_{i=1}^{k} p_i \min_{f \in \mathcal{F}} \varepsilon(f|C_i) \tag{19}$$

Finally, considering the optimal $f'_1, \dots, f'_\ell$ for the partition $C_0, \dots, C_k$ 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 $g(x|s) = \langle \boldsymbol{w}(s), \boldsymbol{\phi}(x) \rangle$ where:

$$\boldsymbol{\phi}(x) = (\tilde{x}_1, \dots, \tilde{x}_\ell)$$
$$\boldsymbol{w}(s) = (w_1(s), \dots, w_\ell(s))$$

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

We can write $\boldsymbol{\phi}(x) = x^\top \Theta$ where $\Theta_{i\cdot} = \theta_i$ are rows, and denote columns by $\bar{\theta}_j = \Theta_{\cdot j}$. Note that:

$$\begin{aligned} g(x|s) &= \sum_{i=1}^{\ell} w_i(s) \langle \theta_i, x \rangle \\ &= \sum_{k=1}^{d} \sum_{i=1}^{\ell} \Theta_{ik} w_i(s) x_k \\ &= \sum_{k=1}^{d} \underbrace{\langle \bar{\theta}_k, \overbrace{w(s) \cdot x_k}^{\text{(I)}} \rangle}_{\text{(II)}} \end{aligned}$$

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).

$$R_f \le X_\infty \sqrt{\frac{2 \log 2d}{m}} \qquad \text{(Lemma 26.11)}$$
$$R_w \le \lambda_w^{(\rho)} R_f \qquad \text{(Lemma 26.9)}$$
$$R_{\text{(I)}} \le \max_x \|x\|_\infty R_w \qquad \text{(Lemma 26.6)}$$
$$R_{\text{(II)}} \le 2 \max_{\bar{\theta}} \|\bar{\theta}\|_1 \cdot R_{\text{(I)}} = 2\|\Theta\|_1 R_{\text{(I)}}$$

where the last inequality follows from Sec. 4 in Sridharan (2014). Assuming $\|\Theta\|_1 \le 1$, combining the above gives:

$$R_g \le 2X_\infty^2 \lambda_w^\rho \sqrt{\frac{2\log 2d}{m}}$$

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

$$\varepsilon(\mathcal{G}) \le \varepsilon_T(\mathcal{G}) + 2R_g + O\left(\sqrt{\frac{\log(1/\delta)}{m}}\right) \tag{20}$$

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

$$r(s) = (r_1(s), \dots, r_\ell(s)), \qquad r_i(s) = r(\tilde{s}_i)$$

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

$$\begin{aligned} R_\mu &\le \lambda_\mu(R_f + R_r) && \text{(Lemma 26.6)} \\ R_r &\le \lambda_r^{(\rho)} && \text{(Lemma 26.6)} \\ R_g &\le 2W_1 R_\mu && \text{(Lemma 26.7)} \end{aligned}$$

and R_f is as before. Together, this gives:

$$R_g \le 2W_1 R_\mu \le 2W_1 \lambda_\mu (1 + \lambda_r^\rho) X_\infty \sqrt{\frac{2\log 2d}{m}}$$

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:
   - Continuous: for some features, a log or square-root transform.
   - Ordinal: One-hot encoding.
   - Date/time: use week and month to capture seasonality of hotel pricing.
   - 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.
   - Additional features: popularity score.[9].

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:

Table 3: Dataset description

Dataset	m	$\lvert\mathcal{X}\rvert$	$\max(n)$	$\text{avg}(n)$	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

- **MNL**: Our implementation.
- **SVMRank**: Open-source code with minor modifications.[11]
- **RankNet**: Learning2rank library.[12]
- **MixedMNL**: Our implementation.
- **AdaRank**: Open-source code.[13].

[9] https://ajourneyintodatascience.quora.com/Learning-to-Rank-Personalize-Expedia-Hotel-Searches-ICDM-2013-Feature-Engineering
[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

- **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 4: Number of parameters

Dataset	SDA	RankNet	Deep Sets
Amadeus	14,144	542,720	201,216
Expedia	7,088	533,504	198,912
Outbrain	8,656	535,552	199,424

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 $F : \mathcal{X} \to \mathbb{R}^k$ and vector-valued reference functions $r : 2^{\mathbb{R}^k} \to \mathbb{R}^k$ for some k. Second, value-reference comparisons within each dimension of aggregation $i \in [\ell]$ are done using an inner product $\langle \tilde{x}, r(\tilde{s}) \rangle$, with multi-dimensional embedded "dimensions" $\tilde{x}_i \in \mathbb{R}^k$ and reference points $r(\tilde{s}_i) \in \mathbb{R}^{\ell \times k}$, $\tilde{s}_i = \{\tilde{x}_i\}_{x \in s}$. 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):

$$\textbf{SDW}: \quad g_{\boldsymbol{f}}(x|s;w) = \sum_{i=1}^{\ell} w(\tilde{s}_i)\tilde{x}_i$$

$$\textbf{SDR}: \quad g_{\boldsymbol{f}}(x|s;v,r,\mu) = \sum_{i=1}^{\ell} v_i \mu\left(\langle \tilde{x}_i, r(\tilde{s}_i)\rangle\right).$$

where $v \in \mathbb{R}^{\ell}$.

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

Table 5: Specification of all ablated models

	ℓ	w	ϕ	r	μ
SDA$_+$	24	Set NN	$\langle \tilde{x}, r(\tilde{s}) \rangle$	Set NN	c-tanh
SDA$_+$, $\mu =$ tanh	24	Set NN	$\langle \tilde{x}, r(\tilde{s}) \rangle$	Set NN	tanh
SDA$_+$, no μ	24	Set NN	$\langle \tilde{x}, r(\tilde{s}) \rangle$	Set NN	-
SDR	24	$\in \mathbb{R}^{\ell}$	$\langle \tilde{x}, r(\tilde{s}) \rangle$	Set NN	c-tanh
SDW	24	Set NN	$\tilde{x}$	-	-
SDW, single f	1	Set NN	$\tilde{x}$	-	-
MNL	1	$= 1$	$\tilde{x}$	-	-

[14] https://github.com/manzilzaheer/DeepSets
[15] https://www.tensorflow.org/
[16] https://optuna.org/

Table 6: Full experimental results.

	Top-1					Top-5					Mean rank				
	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50
SDA_+	**45.42** ± 0.5	**33.48** ± 0.3	**29.26** ± 0.0	**26.57** ± 0.3	**23.23** ± 0.2	**93.37** ± 0.0	**80.40** ± 0.3	**73.77** ± 0.4	**69.64** ± 0.0	**62.35** ± 0.1	**2.31** ± 0.3	**3.50** ± 0.0	**4.33** ± 0.2	**4.93** ± 0.4	**6.37** ± 0.0
MNL (McFadden et al., 1973)	38.42 ± 0.5	27.93 ± 0.4	23.54 ± 0.3	22.31 ± 0.1	18.39 ± 0.2	91.02 ± 0.3	76.51 ± 0.3	68.36 ± 0.4	65.10 ± 0.4	56.20 ± 0.3	2.57 ± 0.0	3.92 ± 0.0	4.94 ± 0.0	5.60 ± 0.1	7.55 ± 0.1
SVMRank (Joachims, 2006)	40.27 ± 0.4	28.17 ± 0.3	23.99 ± 0.3	23.02 ± 0.2	18.64 ± 0.2	91.94 ± 0.3	76.82 ± 0.2	68.56 ± 0.4	66.52 ± 0.2	57.50 ± 0.2	2.49 ± 0.0	3.87 ± 0.0	4.85 ± 0.0	5.35 ± 0.0	7.16 ± 0.0
RankNet (Burges et al., 2005)	37.44 ± 0.7	26.77 ± 0.6	23.81 ± 0.3	20.29 ± 0.7	16.99 ± 0.5	84.67 ± 1.7	66.06 ± 1.9	61.59 ± 0.9	49.45 ± 1.8	44.98 ± 2.6	3.02 ± 0.1	4.98 ± 0.2	5.96 ± 0.1	8.35 ± 0.3	11.07 ± 0.7
Mixed MNL (Train, 2009)	37.96 ± 0.3	27.00 ± 0.2	22.98 ± 0.3	21.68 ± 0.2	17.67 ± 0.3	90.40 ± 0.3	74.80 ± 0.2	65.87 ± 0.4	62.50 ± 0.3	52.87 ± 0.4	2.62 ± 0.0	4.09 ± 0.0	5.26 ± 0.0	6.00 ± 0.1	8.39 ± 0.1
AdaRank (Xu & Li, 2007)	37.27 ± 0.4	25.79 ± 0.3	18.79 ± 0.7	15.79 ± 0.5	11.89 ± 0.2	72.34 ± 0.3	58.28 ± 0.3	51.64 ± 0.6	47.85 ± 1.4	39.08 ± 0.2	4.03 ± 0.0	5.75 ± 0.0	7.14 ± 0.1	8.05 ± 0.3	11.55 ± 0.1
Deep Sets (Zaheer et al., 2017)	40.36 ± 0.5	31.02 ± 0.5	26.66 ± 0.4	25.48 ± 0.3	20.55 ± 0.3	91.92 ± 0.3	79.31 ± 0.2	71.18 ± 0.4	68.76 ± 0.4	59.88 ± 0.4	2.48 ± 0.0	3.64 ± 0.0	4.58 ± 0.0	5.01 ± 0.0	6.75 ± 0.1
Price/Quality	36.44 ± 0.3	25.44 ± 0.2	22.40 ± 0.2	20.26 ± 0.2	16.11 ± 0.1	87.23 ± 0.2	67.77 ± 0.2	58.86 ± 0.2	54.44 ± 0.2	45.43 ± 0.3	2.79 ± 0.0	4.86 ± 0.0	6.32 ± 0.0	7.27 ± 0.0	10.90 ± 0.1
Random	25.15 ± 0.5	14.78 ± 0.2	11.58 ± 0.2	9.91 ± 0.2	6.24 ± 0.2	32.87 ± 0.6	42.60 ± 0.3	34.54 ± 0.2	14.20 ± 0.2	11.04 ± 0.2	6.49 ± 0.0	8.98 ± 0.1	12.03 ± 0.1	18.11 ± 0.1	23.55 ± 0.1

	Expedia												Outbrain		
	Top-1				Top-5				Mean rank				Top-1	Top-5	Mean rank
	10	20	30	40	10	20	30	40	10	20	30	40	12	12	12
SDA_+	**31.49** ± 0.2	**26.81** ± 0.1	**21.96** ± 0.0	**18.36** ± 0.2	**86.91** ± 0.2	73.06 ± 0.0	**61.68** ± 0.2	**53.56** ± 0.4	**2.99** ± 0.0	**4.18** ± 0.2	**5.99** ± 0.2	**7.65** ± 0.0	**38.04** ± 0.3	**94.54** ± 0.1	**2.42** ± 0.0
MNL (McFadden et al., 1973)	30.06 ± 0.2	25.29 ± 0.3	20.61 ± 0.4	16.65 ± 0.4	86.34 ± 0.1	72.96 ± 0.3	60.94 ± 0.5	53.26 ± 0.5	3.07 ± 0.0	4.33 ± 0.0	6.35 ± 0.1	8.48 ± 0.1	37.74 ± 0.3	94.52 ± 0.2	2.43 ± 0.0
SVMRank (Joachims, 2006)	31.28 ± 0.2	26.64 ± 0.3	21.93 ± 0.2	18.03 ± 0.2	86.24 ± 0.1	**73.17** ± 0.3	60.53 ± 0.2	52.25 ± 0.2	3.01 ± 0.0	4.19 ± 0.0	6.12 ± 0.0	7.95 ± 0.0	37.68 ± 0.3	94.46 ± 0.1	2.43 ± 0.0
RankNet (Burges et al., 2005)	23.82 ± 0.5	18.60 ± 0.7	11.48 ± 0.6	11.54 ± 0.4	81.85 ± 0.6	62.91 ± 0.9	43.09 ± 1.1	38.49 ± 0.9	3.43 ± 0.0	5.21 ± 0.1	8.72 ± 0.2	10.49 ± 0.2	35.32 ± 0.8	91.55 ± 0.7	2.65 ± 0.1
Mixed MNL (Train, 2009)	27.28 ± 0.6	22.00 ± 0.7	18.32 ± 0.4	13.91 ± 0.6	84.24 ± 0.3	68.31 ± 0.7	55.41 ± 0.6	43.55 ± 1.0	3.22 ± 0.0	4.67 ± 0.1	6.81 ± 0.1	9.45 ± 0.2	37.72 ± 0.3	94.42 ± 0.1	2.43 ± 0.0
AdaRank (Xu & Li, 2007)	26.70 ± 0.2	22.57 ± 0.3	17.47 ± 0.2	14.14 ± 0.2	83.21 ± 0.2	68.14 ± 0.3	54.46 ± 0.3	44.89 ± 0.2	3.29 ± 0.0	4.71 ± 0.0	7.01 ± 0.0	9.33 ± 0.0	37.47 ± 0.3	94.40 ± 0.2	2.44 ± 0.0
Deep Sets (Zaheer et al., 2017)	29.87 ± 0.3	25.64 ± 0.2	20.95 ± 0.3	16.74 ± 0.3	86.26 ± 0.2	72.55 ± 0.3	60.25 ± 0.3	51.35 ± 0.3	3.06 ± 0.0	4.26 ± 0.0	6.19 ± 0.0	8.08 ± 0.1	37.51 ± 0.3	94.30 ± 0.1	2.44 ± 0.0
Price/Quality	17.92 ± 0.1	13.24 ± 0.2	9.92 ± 0.1	7.80 ± 0.1	77.67 ± 0.1	56.00 ± 0.2	42.50 ± 0.2	32.94 ± 0.3	3.79 ± 0.0	5.84 ± 0.0	8.60 ± 0.0	11.21 ± 0.1	24.17 ± 0.1	25.08 ± 0.1	8.13 ± 0.0
Random	14.13 ± 0.1	9.68 ± 0.2	6.89 ± 0.1	5.05 ± 0.1	32.35 ± 0.2	33.44 ± 0.2	24.16 ± 0.2	13.25 ± 0.2	6.38 ± 0.0	8.65 ± 0.0	12.01 ± 0.0	17.89 ± 0.1	22.21 ± 0.1	23.10 ± 0.1	8.32 ± 0.0

Table 7: Ablation Experiment Result.

	Amadeus														
	Top-1					Top-5					Mean rank				
	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50
SDA_+	**45.42** ±0.5	**33.48** ±0.3	**29.26** ±0.0	26.57 ±0.3	**23.23** ±0.2	**93.37** ±0.0	**80.40** ±0.3	**73.77** ±0.4	69.64 ±0.0	**62.35** ±0.1	**2.31** ±0.3	**3.50** ±0.0	**4.33** ±0.2	4.93 ±0.4	**6.37** ±0.0
SDA_+ with $\mu = \tanh$	39.10 ±0.4	31.54 ±0.3	27.02 ±0.4	25.98 ±0.2	20.67 ±0.3	91.40 ±0.3	79.12 ±0.2	71.05 ±0.3	68.85 ±0.2	59.50 ±0.2	2.55 ±0.0	3.64 ±0.0	4.60 ±0.0	5.04 ±0.0	6.85 ±0.0
SDA_+ with no μ	41.38 ±0.5	33.26 ±0.3	28.89 ±0.3	27.28 ±0.2	22.16 ±0.2	92.19 ±0.3	80.22 ±0.3	73.08 ±0.4	**69.92** ±0.3	61.19 ±0.4	2.45 ±0.0	3.53 ±0.0	4.39 ±0.0	**4.87** ±0.0	6.56 ±0.0
SDR	37.50 ±0.3	30.64 ±0.3	26.16 ±0.3	24.76 ±0.2	15.81 ±0.2	87.31 ±0.3	76.53 ±0.2	69.76 ±0.4	66.67 ±0.2	46.81 ±0.3	2.82 ±0.0	3.99 ±0.0	4.92 ±0.0	5.49 ±0.0	12.23 ±0.0
SDW	39.98 ±0.4	32.03 ±0.3	27.85 ±0.3	26.29 ±0.3	20.15 ±0.2	91.89 ±0.3	79.57 ±0.2	71.82 ±0.3	68.62 ±0.3	58.55 ±0.2	2.50 ±0.0	3.59 ±0.0	4.51 ±0.0	5.04 ±0.0	6.97 ±0.0
SDW with single f	38.21 ±0.4	27.60 ±0.3	23.66 ±0.4	22.20 ±0.1	18.41 ±0.3	90.90 ±0.4	75.49 ±0.3	66.46 ±0.4	63.28 ±0.2	54.11 ±0.4	2.59 ±0.0	4.00 ±0.0	5.20 ±0.1	5.87 ±0.0	8.26 ±0.1
MNL	38.42 ±0.5	27.93 ±0.4	23.54 ±0.3	22.31 ±0.1	18.39 ±0.2	91.02 ±0.3	76.51 ±0.3	68.36 ±0.4	65.10 ±0.4	56.20 ±0.3	2.57 ±0.0	3.92 ±0.0	4.94 ±0.0	5.60 ±0.1	7.55 ±0.1

	Expedia												Outbrain		
	Top-1				Top-5				Mean rank				Top-1	Top-5	Mean rank
	10	20	30	40	10	20	30	40	10	20	30	40	12	12	12
SDA_+	**31.49** ±0.2	26.81 ±0.1	**21.96** ±0.0	**18.36** ±0.2	**86.91** ±0.2	73.06 ±0.0	**61.68** ±0.2	**53.56** ±0.4	**2.99** ±0.0	4.18 ±0.2	**5.99** ±0.2	**7.65** ±0.0	38.04 ±0.3	94.54 ±0.1	**2.42** ±0.0
SDA_+ with $\mu = \tanh$	31.19 ±0.1	26.26 ±0.2	21.81 ±0.2	18.10 ±0.2	86.18 ±0.2	72.61 ±0.3	60.90 ±0.3	52.91 ±0.2	3.02 ±0.0	4.24 ±0.0	6.08 ±0.0	7.81 ±0.0	**38.05** ±0.3	**94.56** ±0.2	2.42 ±0.0
SDA_+ with no μ	30.90 ±0.2	26.52 ±0.2	21.88 ±0.2	18.20 ±0.2	86.31 ±0.2	72.80 ±0.2	60.79 ±0.4	52.96 ±0.2	3.02 ±0.0	4.21 ±0.0	6.08 ±0.0	7.73 ±0.0	37.79 ±0.3	94.48 ±0.1	2.43 ±0.0
SDR	25.05 ±0.2	22.43 ±0.2	18.24 ±0.2	17.63 ±0.1	80.67 ±0.2	68.61 ±0.2	55.40 ±0.3	51.19 ±0.3	3.46 ±0.0	4.75 ±0.0	7.04 ±0.0	8.11 ±0.0	37.98 ±0.3	94.54 ±0.2	2.42 ±0.0
SDW	31.27 ±0.2	26.80 ±0.2	21.88 ±0.2	18.22 ±0.2	86.67 ±0.2	**73.08** ±0.2	61.07 ±0.3	53.41 ±0.2	2.99 ±0.0	**4.17** ±0.0	6.05 ±0.0	7.73 ±0.0	37.86 ±0.3	94.48 ±0.2	2.43 ±0.0
SDW with single f	30.04 ±0.2	25.28 ±0.3	20.82 ±0.3	16.64 ±0.4	85.41 ±0.1	71.88 ±0.3	58.98 ±0.5	49.54 ±0.5	3.07 ±0.0	4.34 ±0.0	6.32 ±0.1	8.48 ±0.1	37.80 ±0.3	94.52 ±0.1	2.43 ±0.0
MNL	30.06 ±0.2	25.29 ±0.3	20.61 ±0.4	16.65 ±0.4	86.34 ±0.1	72.96 ±0.3	60.94 ±0.5	53.26 ±0.5	3.07 ±0.0	4.33 ±0.0	6.35 ±0.1	8.48 ±0.1	37.74 ±0.3	94.52 ±0.2	2.43 ±0.0