DECLARATIVE CHARACTERIZATIONS OF DIRECT PREFERENCE ALIGNMENT ALGORITHMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent direct preference alignment algorithms (DPA), such as DPO, have shown great promise in aligning large language models to human preferences. While this has motivated the development of many new variants of the original DPO loss, understanding the differences between these recent proposals, as well as developing new DPA loss functions, remains difficult given the lack of a technical and conceptual framework for reasoning about the underlying semantics of these algorithms. In this paper, we attempt to remedy this by formalizing DPA losses in terms of discrete reasoning problems. Specifically, we ask: *Given an existing DPA loss, can we systematically derive a symbolic expression that characterizes its semantics? How do the semantics of two losses relate to each other?* We propose a novel formalism for characterizing preference losses for single model and reference model based approaches, and identify symbolic forms for a number of commonly used DPA variants. Further, we show how this formal view of preference learning sheds new light on both the size and structure of the DPA loss landscape, making it possible to not only rigorously characterize the relationships between recent loss proposals but also to systematically explore the landscape and derive new loss functions from first principles. We hope our framework and findings will help provide useful guidance to those working on human AI alignment.

1 INTRODUCTION

Symbolic logic has long served as the de-facto language for expressing complex knowledge throughout computer science (Halpern et al., 2001), including in artificial intelligence (McCarthy et al., 1960; Nilsson, 1991), owing to its declarative nature and clean semantics. Symbolic approaches to reasoning that are driven by declarative knowledge, in sharp contrast to purely machine learning-based approaches, have the advantage of allowing us to reason transparently about the behavior and correctness of the resulting systems. In this paper we focus on the broad question: *Can the declarative modeling approach be used to better understand and formally specify learning algorithms for large language models (LLMs)?*

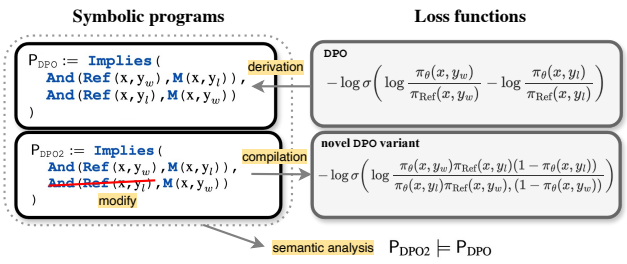

Figure 1: Can we uncover the hidden logic of DPO? Here we show the distillation of the DPO loss to a symbolic expression that expresses its high-level model behavior, along with a modified version of that program that we can compile into a novel DPO loss.

We specifically investigate direct preference learning algorithms, such as direct preference optimization (DPO) (Rafailov et al., 2024), for pairwise preference learning, which are currently at the forefront of research on LLM alignment and learning from human preferences (Ouyang et al., 2022; Wang et al., 2023). While there has been much recent work on algorithmic variations of DPO (Azar et al., 2023; Hong et al., 2024; Meng et al., 2024, *inter alia*) that modify or add new terms to the original loss, understanding the differences between these new proposals, as

well as coming up with new variants, remains a formidable challenge due to the lack of a conceptual and technical framework for reasoning about their underlying semantics.

Our study attempts to remedy this problem by formalizing the corresponding loss functions in terms of logic. Such a formalization is based on trying to answer the following question: *Given an existing loss function, such as DPO (see Figure 1), can we derive a symbolic expression that captures the core semantics of that loss function (i.e., one that we can then systematically compile back into the exact loss)?* In treating loss functions as discrete reasoning problems, ones that abstract away from certain lower-level details about optimization and tell us about high-level model behavior, it becomes possible to study them using conventional semantic notions from logic and probability (e.g., *logical entailment*), relate it semantically to other programs, or even modify its underlying logical semantics to derive entirely new algorithms.

To facilitate this formalization, we devise a novel probabilistic logic based on a generalization of the notion of *semantic loss* (SL) Xu et al. (2018) coupled with a provably correct mechanical procedure for translating existing DPA losses into programs in our logic. As in SL, losses are produced from symbolic programs by counting the weighted propositional models of those programs, reducing the problem to one of standard probabilistic inference (Chavira & Darwiche, 2008). In contrast to the kinds of symbolic programs commonly used with SL, however, empirically successful DPA losses impose systematic conditional constraints on the types of models that should be counted, which shape the structure of the underlying probability distribution. We express these constraints through a new primitive in our logic called a **preference structure** that also addresses various technical and conceptual issues involved with modeling pairwise preference symbolically. It is through such constraints that certain semantic relationships between existing losses can be easily observed and new losses can be derived.

Our formal view of preference learning sheds much light on the size and structure of the **DPA loss landscape**. Under modest assumptions motivated by the structure of existing DPA losses and our new logic, we see that the number of definable DPA losses is doubly exponential over the number (n) of unique predictions (i.e., forward model calls) made in a loss function, or 4^{2^n}. This results in, for example, close to 4.3 billion unique variations of the original DPO loss. While big, we show how this space is structured in interesting ways based on formal connections between relationships that hold in the semantic space among formalized DPA losses (e.g., logical entailment, equivalence) and their monotonicity properties in the loss space.

These formal results also provide practical insights into how to effectively search for new DPA losses. For example, one can start with empirically successful loss functions, use the formalization to understand their semantics, then modify their semantics to arrive at novel variants that are either more constrained or less, then experiment accordingly.

2 Related work

Language model alignment While traditional approaches to language model alignment have employed reinforcement learning (Ziegler et al., 2019; Christiano et al., 2017), we focus on DPA approaches such as `DPO` (Rafailov et al., 2024) and `SliC` (Zhao et al., 2023) that use closed-form loss functions to tune models directly to offline preferences.

We touch on two recent areas in this space: formal characterizations of DPA losses (Azar et al., 2023; Tang et al., 2024; Hu et al., 2024) and work on devising algorithmically enhanced variants of `DPO` (Amini et al., 2024; Hong et al., 2024; Meng et al., 2024; Pal et al., 2024; Xu et al., 2024; Ethayarajh et al., 2024; Park et al., 2024). In contrast to this work on formal characterization, which focuses on the optimization properties of DPA losses and particular parameterizations like Bradley-Terry, we attempt to formally characterize the semantic relationships between these variants of `DPO` in an optimization agnostic way to better understand the structure of the DPA loss landscape.

Neuro-symbolic modeling For formalization, we take inspiration from work on compiling symbolic formulas into novel loss functions (Li et al., 2019; Fischer et al., 2019; Marra et al., 2019; Asai & Hajishirzi, 2020, *inter alia*), which is used for incorporating background constraints into learning that have shown to improve training robustness and model consistency. In particular, we focus on approaches based on probabilistic logic (Xu et al., 2018; Manhaeve et al., 2018; Ahmed et al., 2022; 2023; van Krieken et al., 2024b).

In contrast to this work, however, we focus on the inverse problem of **decompilation**, or deriving symbolic expressions from known and empirically successful loss functions to better understand their semantics (see Friedman et al. (2024) for a similar idea related to decompiling LLMs). To our knowledge, work in this area has mostly been limited to symbolically deriving standard loss function such as cross-entropy (Giannini et al., 2020; Li et al., 2019), whereas we look at more complex preference learning loss functions at the forefront of LLM research.

Language model programming Finally, we take inspiration from recent work on formalizing LLM algorithms in terms of programming language concepts (Dohan et al., 2022; Beurer-Kellner et al., 2023; Khattab et al., 2023), with our approach being declarative in style. As such, our study relates to work on declarative programming techniques for ML (Eisner et al., 2004; De Raedt et al., 2007; De Raedt & Kimmig, 2015; Li et al., 2023; Vieira et al., 2017; van Krieken et al., 2024a).

3 Direct Preference Alignment

In this section, we review the basics of offline preference alignment, which can be defined as the following problem: given data of the form: $D_{\text{p}} = \left\{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\right\}_{i=1}^{M}$ consisting of a model input x and two possible generation outputs (often ones rated by humans), a preferred output y_w (the *winner* w) and a dispreferred output y_l (the *loser* l), the goal is to optimize a policy model (e.g., an LLM) $\pi_\theta(\cdot \mid x)$ to such preferences.

As mentioned at the outset, we focus on direct preference alignment (DPA) approaches that all take the form of some closed-form loss function ℓ that we can use to directly train our model on D_{p} to approximate the corresponding ground preference distribution $p^*(y_w \succ y_l \mid x)$. Since our study focuses on the formal properties of DPA losses, it is important to understand their general structure, which will take the following form (Tang et al., 2024):

	$f(\rho_\theta, \beta) =$	ρ_θ (standard formulation)
`DPO`	$-\log \sigma(\beta \rho_\theta)$	$\log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}$
`IPO`	$(\rho_\theta - \frac{1}{2\beta})^2$	
`SliC`	$\max(0, \beta - \rho_\theta)$	$\log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}$
`RRHF`	$\max(0, -\rho_\theta)$	$\log \frac{\exp\left(\frac{1}{\lvert y_w \rvert} \log \pi_\theta(y_w \mid x)\right)}{\exp\left(\frac{1}{\lvert y_l \rvert} \log \pi_\theta(y_l \mid x)\right)}$

Table 1: Examples of some popular DPA loss functions with different choices of f and ρ_θ.

$$\ell_{\text{DPA}}(\theta, D) := \mathop{\mathbb{E}}_{(x, y_w, y_l) \sim D_{\text{p}}} \left[f\big(\rho_\theta(x, y_w, y_l), \beta\big) \right] \tag{1}$$

consisting of some convex loss function $f : \mathbb{R} \times \mathbb{R}+ \to \mathbb{R}$, a model quantity $\rho_\theta(x, y_w, y_l)$ which we will abbreviate to ρ_θ and a parameter β.[1]

Table 1 lists four specific DPA losses: `DPO` (Rafailov et al., 2024), `IPO` (Azar et al., 2023), `SliC` (Zhao et al., 2022; 2023), and `RRHF` (Yuan et al., 2023). Here the logistic log loss (shown using the logistic function $\sigma(x) = \frac{1}{1+\exp(-x)}$), square loss, hindge loss, and perceptron loss are used for f, respectively. Loss functions such as `SliC` and `RRHF` are examples of single model approaches define ρ_θ in terms of the **log ratio of the winner and loser** given prediction probabilities π_θ of the model being trained. As an important implementation detail, the prediction probabilities are sometimes computed using log length normalization as shown for `RRHF`. For `DPO` and `IPO`, in contrast, the model quantity ρ_θ is the **log ratio difference** (of the winner and the loser) between the predictions of the model being trained and a frozen LLM called a reference model, π_{ref}. These two approaches constitute a two model approach, where the role of the reference model is to avoid overfitting on the target preference data (controlled by the parameter β).

Single model approaches have the advantage of avoiding the overhead associated with having an additional reference model and can sometimes yield competitive performance when compared against two model approaches. In the absence of a reference model, these losses are usually regularized using an added cross-entropy term, which we exclude from our current analysis.[2]

[1] Following Tang et al. (2024) and their GPO framework, we formulate DPA approaches as general binary classification problems and do not make any assumptions about the preference structure $p(y_w \succ y_l \mid x)$.

[2] When referring to the `CPO`, `ORPO` and `SliC` losses, we refer to the losses without the cross-entropy terms. For example, what we call `SliC` and `ODPO` refers to the `cal` and `OR` losses, respectively, in the original papers.

The structure of DPA variants. Conceptually, preference losses involve making predictions about winners and losers across models and reasoning about the relationships between predictions. The main question we ask is: *If we view this process as a discrete reasoning problem, what is the nature of the reasoning that underlies these different losses and each ρ_θ?* To do our analysis, we start by rewriting each loss function in a way that strips away various optimization and implementation details (e.g., details about f, β and the choice about whether length normalization is used) in order to arrive at a bare form of ρ_θ.

Accordingly, we will write $P_m(y \mid x)$ in place of $\pi_\theta(y \mid x)$ to denote the probability assigned by a model m to an output y in a way that is agnostic to whether length normalization is used. In Table 2, we show different variants of DPO that we consider and two common baselines, the cross-entropy loss $\ell_{\texttt{CE}}$ an a variant that uses an unlikelihood (Welleck et al., 2019) term ℓ_{CEUnl}. Importantly, we later express each ρ_θ as a single log ratio $\rho^t_\theta/\rho^b_\theta$, which we refer to as the **core loss equation** for each individual loss.

Loss	$\rho_\theta := \log \frac{\rho^t_\theta}{\rho^b_\theta}$		$s_{m_1,m_2}(y_1,y_2) := \log \frac{P_{m_1}(y_1\|x)}{P_{m_2}(y_2\|x)}$
Baselines ρ_θ			
$\ell_{\texttt{CE}}$	$\log \frac{P_\theta(y_w\|x)}{1-P_\theta(y_w\|x)}$	ℓ_{CEUnl}	$\log \frac{P_\theta(y_w\|x)(1-P_\theta(y_l\|x))}{P_\theta(y_l\|x)+(1-P_\theta(y_w\|x)))}$
Single model approaches (no reference) P_θ			
$\ell_{\texttt{CPO}}$	$\log \frac{P_\theta(y_w\|x)}{P_\theta(y_l\|x)}$	$s_\theta(y_w,y_l)$	
$\ell_{\texttt{ORPO}}$	$\log \frac{P_\theta(y_w\|x)(1-P_\theta(y_l\|x))}{P_\theta(y_l\|x)(1-P_\theta(y_w\|x))}$	$s_\theta(y_w,y_l)$	$-s_\theta(\overline{y_w},\overline{y_l})$
$\ell_{\texttt{SimPO}}$	$\log \frac{P_\theta(y_w\|x)P_{\text{mref}}(y_l\|x)}{P_{\text{mref}}(y_w\|x)P_\theta(y_l\|x)}$	$s_\theta(y_w,y_l)$	$-s_{\text{mref}}(y_w,y_l)$
with reference model P_{ref}			
$\ell_{\texttt{DPO}}$	$\log \frac{P_\theta(y_w\|x)P_{\text{ref}}(y_l\|x)}{P_{\text{ref}}(y_w\|x)P_\theta(y_l\|x)}$	$s_\theta(y_w,y_l)$	$-s_{\text{ref}}(y_w,y_l)$
$\ell_{\texttt{DPOP}}$	$\log \frac{P_\theta(y_w\|x)P_{\theta 2}(y_w\|x)P_{\text{ref}}(y_l\|x)}{P_{\text{ref}}(y_w\|x)P_{\text{ref2}}(y_w\|x)P_\theta(y_l\|x)}$	$s_\theta(y_w,y_l)$	$-s_{\text{ref}}(y_w,y_l)$ $-s_{\text{ref2},\theta 2}(y_w,y_w)$

Table 2: How are variants of DPO structured? Here we define some popular variants in terms of their **core loss equation** ρ_θ and the helper function $s_{m_1,m_2}(y_1,y_2)$ (last column) that rewrites each ρ_θ in a way that brings out general shared structural patterns and added terms compared with the log win/loss ratio $s_\theta(y_w,y_l)$.

To more easily see the relationships between these proposals, we rewrite each ρ_θ in terms of the log ratio function $s_m(y_1,y_2)$ defined in Table 2 (we use $\overline{y}$ to denote the negation of y, or $1 - P_m(y \mid x)$). Here we see that all losses are derivable from the log ratio of winner and loser $s_\theta(y_w,y_l)$ used in SliC and RRHF either exactly, as in CPO (Xu et al., 2024), or with added terms. DPO, for example, is expressible as this ratio minus an additional log ratio term $s_{\text{ref}}(y_w,y_l)$ that contains information about the reference model. Many variations to DPO then involve making the following two modifications.

Adding additional terms. Approaches like $\ell_{\texttt{DPOP}}$ (Pal et al., 2024) (see also Amini et al. (2024); Park et al. (2024)) incorporate additional terms into DPO ($s_{\text{ref2},\theta 2}(y_w,y_w)$) that address particular failure cases. We use $\theta 2$ and ref2 to refer to copies of our two models, which is a decision that we address later when discussing the structure of the equation class ρ_θ.

Changing the reference ratio. Approaches, such as $\ell_{\texttt{ORPO}}$ Hong et al. (2024) and $\ell_{\texttt{SimPO}}$ Meng et al. (2024) instead reparameterize the reference ratio $s_{\text{ref}}(y_w,y_l)$ either in terms of some quantity from our policy model as in ORPO ($s_\theta(\overline{y_w},\overline{y_l})$) or a heuristic penalty term γ as in SimPO. For SimPO rewrite their γ term in terms of the log ratio $\gamma = s_{\text{mref}}(y_w,y_l)$ (where 'mref' refers to a *manual* approximation of the reference model) to make it align to the structure of DPO.

4 Preference Modeling as a Reasoning Problem

To better understand the DPA loss space, we will formalize the preference losses and the model quantities ρ_θ introduced in the previous section in terms of symbolic reasoning problems. This will involve the following core ideas and assumptions.

Model predictions are symbolic objects The declarative approach will involve thinking of LLMs predictions as logical propositions. For example, when a model M generates an output y_w for a prompt x, we will use the notation $\texttt{M}(x,y_w)$ to express the proposition that y_w *is a valid generation for* x. Importantly, we will further weight these propositions by assigning the probabilities given by the corresponding model, i.e., $P_\theta(\texttt{M}(x,y_w)) = P_\theta(y_w \mid x)$. We call these propositions our **probabilistic predictions** $X_1,...,X_n$, which will form the basis of symbolic formulas.

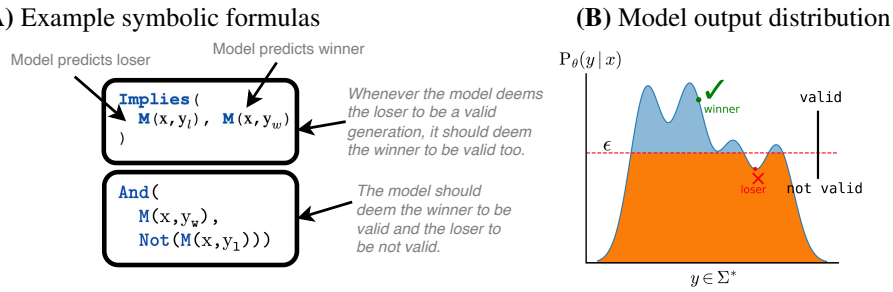

Figure 2: What do formal representations of loss functions tell us? We show (A) two symbolic formulas related to single model preference learning with their semantics paraphrased in informal English. When grounded in model behavior, they tell us about the structure of the model's output probability distribution (B) and where predictions belong in that distribution (relative to some ϵ).

Relationships between predictions are expressed as symbolic formulas Relationships between model predictions will take the form of symbolic constraints expressed as formulas of propositional logic P defined by applying zero or more Boolean operators over probabilistic predictions. For example, in Figure 2 (A), the top formula uses the implication operator (`Implies`) to express the constraint that model `M` should never deem the loser y_l to be a valid generation (`M`(x, y_l)) without deeming the winner y_w to also be valid (`M`(x, y_w)). The bottom formula, in contrast, tells us that only the winner y_w should be deemed valid using the conjunction and negation operators (`And`, `Not`).[3]

When grounded to model behavior via the proposition weights, such constraints tell us about the structure of a model's output probability distribution, as visualized in Figure 2 (B). Semantically, we assume that what constitutes a valid generation is any probabilistic prediction whose weight exceeds some threshold ϵ in that distribution, similar to the notion of ϵ-truncated support from Hewitt et al. (2020). While our results later will not depend on making any direct assumptions about ϵ, such a definition is merely meant to provide intuitions for how to understand our formulas.

Loss functions are expressible as symbolic formulas We assume that all preference loss functions have an internal logic that can be expressed in the form described above. Our main goal is to uncover that internal logic, and to use semantic concepts, such as entailment (denoted as $\models$) or logical equivalence ($\equiv$) to meaningfully characterize the DPA loss space.

4.1 Compilation and Decompilation

Compilation and semantic loss To compile a symbolic formula P into loss, we employ a probabilistic approach based on the semantics of weighted model counting (WMC) (Chavira & Darwiche, 2008; Fierens et al., 2015). This is based on computing a probability of a formula P, given by

$$p_\theta(\mathsf{P}) = \mathrm{WMC}\big(\mathsf{P}; \theta\big) := \sum_{\mathbf{w} \in \{0,1\}^n} \mathbb{1}\{\mathbf{w} \models \mathsf{P}\} \prod_{\mathbf{w} \models X_i} P_\theta(X_i) \cdot \prod_{\mathbf{w} \models \neg X_i} \big(1 - P_\theta(X_i)\big) \tag{2}$$

or as a weighted sum over all the propositional models of that formula $\mathbf{w} \models \mathsf{P}$, or truth assignments where P is satisfied. Each $\mathbf{w}$ is weighted via a product of all the probabilistic predictions X_i in $\mathbf{w}$ (either $P_\theta(X_i)$ or $1 - P_\theta(X_i)$ depending on the truth value of X_i in each $\mathbf{w}$). A loss can then be obtained by taking the negative logarithm of this probability, which is known as the semantic loss first defined in Xu et al. (2018).

Formally, the semantic loss takes the form $\mathbb{E}_{d \sim D}[-\log p_\theta\big(\mathsf{P}_d\big)]$, where we use the notation P_d to refer to the substitution of variables in our formulas P (e.g., x, y_w, y_l) with specific values from $d \sim D$. Since our approach will later involve computing the probability of P conditioned (optionally) on some **conditioning constraints** $\mathsf{P_C}$ (i.e., an additional propositional formula), we consider the

[3]We will switch between using conventional logical notation (e.g., $\wedge, \vee, \neg, \rightarrow$) and operator notation (e.g., `And`, `Or`, `Not`, `Implies`) depending on the context.

conditional form of the semantic loss and its full objective below:

$$\min_{\theta} \mathop{\mathbb{E}}_{d\sim D}\left[-\log p_\theta(\mathsf{P}_d \mid \mathsf{P}_{\mathbf{C}_d})\right], \;\; p_\theta(\mathsf{P} \mid \mathsf{P}_{\mathbf{C}}) = \frac{\text{WMC}\big(\mathsf{P}\wedge\mathsf{P}_{\mathbf{C}};\theta\big)}{\text{WMC}\big(\mathsf{P}\wedge\mathsf{P}_{\mathbf{C}};\theta\big)+\text{WMC}\big(\neg\mathsf{P}\wedge\mathsf{P}_{\mathbf{C}};\theta\big)} \tag{3}$$

where the last part follows from the standard definition of conditional probability (with the denominator being an expanded form of $\text{WMC}\big(\mathsf{P}_{\mathbf{C}};\theta\big)$). We note that when $\mathsf{P}_{\mathbf{C}}$ is equal to $\top$ (or true), this form of the semantic loss is equivalent to the original version.

As an important technical point, we see below how having an explicit negation $\neg\mathsf{P}$ in the normalization allows us write the probability of P in the following way (without loss of generality, we exclude $\mathsf{P}_{\mathbf{C}}$ to improve readability and remove θ from WMC):

$$p_\theta(\mathsf{P}) = \frac{\exp\big(\log \text{WMC}(\mathsf{P})\big)}{\exp\big(\log \text{WMC}(\mathsf{P})\big)+\exp\big(\log \text{WMC}(\neg\mathsf{P})\big)} = \sigma\Bigg(\underbrace{\log\frac{\text{WMC}(\mathsf{P})}{\text{WMC}(\neg\mathsf{P})}}_{\textbf{semantic loss ratio } \rho_{\text{sem}}}\Bigg) \tag{4}$$

$$\text{with } \ell(\mathsf{P},\theta,D) := \mathop{\mathbb{E}}_{d\sim D}\Big[-\log\sigma\Big(\rho_{\text{sem}}(d)\Big)\Big] \tag{5}$$

yielding a logistic log form of the semantic loss $\ell(\mathsf{P},\theta,D)$ that aligns with the structure of the DPA losses in Section 3, where, as an analog to ρ_θ, we call the inner part the **semantic loss ratio** ρ_{sem}.

Decompilation The goal of decompilation is to derive for a loss function ℓ_x a symbolic expression P that characterizes the semantics of that loss. As we show later in Sec. 5.2, this will reduce to the problem of finding a program whose *semantic loss ratio* is equivalent to a loss's *core loss equation* ρ_θ, based largely on the derivation above and its connection with DPA.

5 A LOGIC FOR PREFERENCE MODELING

In the standard semantic loss (SL), ML loss functions ℓ_x are expressible as a single propositional formulas P interpreted via probabilistic logic, with $\ell_x \sim -\log p_\theta(\mathsf{P})$. At first glance, this formulation is at odds with standard formulations of pairwise preference, such as the Bradley-Terry (BT) model (Bradley & Terry, 1952) typically assumed in RLHF, which involves modeling a preference distribution $p_\theta(y_w \succ y_l)$ between two (often disparate) quantities (e.g., given by the kinds of log ratios in Table 2). Indeed, logical accounts of pairwise preference such as Rescher (1967) (see Cvetković (1993)) assume a similar semantics where preference is defined not as a single propositional formula but as an inequality between the counts of two independent formulas $\text{WMC}(\mathsf{P}_w) > \text{WMC}(\mathsf{P}_l)$.

As it turns out, none of the variations of `DPO` and their log ratios in Table 2 can be expressed as a single formula in standard SL.[4] While this can be remedied by modifying the SL to involve counting multiple formulas as in Rescher (1967), we instead define a relational structure called a preference structure that allows us to capture the semantics of losses in a modular fashion using a single propositional formula coupled with auxiliary constraints. Such a structure, which is based on a novel construction in propositional logic, will later make it easy to cleanly characterize different DPA losses and devise new variants through manipulation to their constraints.

Preference structure A preference structure is a tuple $\overline{\mathsf{P}} = (\mathsf{P}, \mathsf{P}_{\mathbf{C}}, \mathsf{P}_{\mathbf{A}})$ consisting of three propositional formulas: a **core semantic formula** P coupled with **conditioning constraints** $\mathsf{P}_{\mathbf{C}}$ (as in Eq 3, which restrict the propositional models that can be counted) and **additive constraints** $\mathsf{P}_{\mathbf{A}}$ that tell us what propositional models always need to be counted. As we will show, all the DPA losses in Table 2 are representable as preference structures, often ones where the same core formula P is shared (e.g., the formulas in Figure 2), yet that differ in the constraints they impose ($\mathsf{P}_{\mathbf{C}}$ and $\mathsf{P}_{\mathbf{A}}$).

[4] To see this for the ratio $s_\theta(y_w, y_l)$ from Table 2, one can enumerate all 16 unique Boolean functions over variables y_w and y_l to see that none yield a semantic formula whose WMC is equal to $\sigma(s_\theta(y_w, y_l))$. Through further analysis, one can also see that it is not possible to derive $\sigma(s_\theta(y_w, y_l))$ using conditional WMC.

Each preference structure will have a **formula form** $\overline{\mathsf{P}_f}$ and a **negated formula form** $\overline{\neg\mathsf{P}_f}$, which are defined by the following two propositional formulas:

$$\overline{\mathsf{P}_f} := \Big(\mathsf{P} \vee \mathsf{P}_{\mathbf{A}}\Big) \wedge \mathsf{P}_{\mathbf{C}}, \quad \overline{\neg\mathsf{P}}_f := \Big(\neg\mathsf{P} \vee \mathsf{P}_{\mathbf{A}}\Big) \wedge \mathsf{P}_{\mathbf{C}}. \tag{6}$$

In the absence of the additive constraint $\mathsf{P}_{\mathbf{A}}$, we note that these representations encode the conditional $\mathsf{P} \mid \mathsf{P}_{\mathbf{C}}$, thus making the semantic loss of these formulas equivalent to the conditional semantic loss in Eq 3. Indeed, many DPA losses will be reducible to the conditional semantic loss, however, $\mathsf{P}_{\mathbf{A}}$ and the ability to add default model counts to P and $\neg\mathsf{P}$ will be needed to derive some DPA losses symbolically and account for peculiar properties of their normalization.

Below we show that any two propositional formulas can be expressed as a preference structure based on a particular construction, called the implication form, that we use later for decompilation.

Proposition 1. *Given any two propositional formulas* P_1 *and* P_2*, there exists a preference structure* $\overline{\mathsf{P}}$ *such that* $\mathsf{P}_1 \equiv \overline{\mathsf{P}_f}$ *and* $\mathsf{P}_2 \equiv \overline{\neg\mathsf{P}_f}$.

Proof. We provide a specific construction we call the **implication form** of P_1 and P_2. This is based on the following logical equivalences (the correctness of which can be checked manually):

$$\mathsf{P}_1 \equiv \Bigg(\underbrace{(\mathsf{P}_2 \rightarrow \mathsf{P}_1)}_{\mathsf{P}} \vee \underbrace{(\mathsf{P}_1 \wedge \mathsf{P}_2)}_{\mathsf{P}_{\mathbf{A}}}\Bigg) \wedge \underbrace{(\mathsf{P}_1 \vee \mathsf{P}_2)}_{\mathsf{P}_{\mathbf{C}}}, \mathsf{P}_2 \equiv \Bigg(\underbrace{\neg(\mathsf{P}_2 \rightarrow \mathsf{P}_1)}_{\neg\mathsf{P}} \vee \underbrace{(\mathsf{P}_1 \wedge \mathsf{P}_2)}_{\mathsf{P}_{\mathbf{A}}}\Bigg) \wedge \underbrace{(\mathsf{P}_1 \vee \mathsf{P}_2)}_{\mathsf{P}_{\mathbf{C}}}$$

As noted above, this construction corresponds exactly to the preference structure $(\mathsf{P}, \mathsf{P}_{\mathbf{C}}, \mathsf{P}_{\mathbf{A}})$ with $\mathsf{P} := \mathsf{P}_2 \rightarrow \mathsf{P}_1$, $\mathsf{P}_{\mathbf{C}} := \mathsf{P}_1 \vee \mathsf{P}_2$ and $\mathsf{P}_{\mathbf{A}} := \mathsf{P}_1 \wedge \mathsf{P}_2$ and its two formula forms. (As a special case, whenever $\mathsf{P}_2 \equiv \neg\mathsf{P}_1$, this simplifies to the structure $\overline{\mathsf{P}} = (\mathsf{P}_1, \top, \bot)$, thus making any single formula representable as a preference structure.) □

5.1 Generalized semantic loss based on preference structures

In our generalization of the semantic loss, formulas P will be replaced with preference structures $\overline{\mathsf{P}}$. For example, we can modify the logistic log form of SL in Eq 5 to be $\ell(\overline{\mathsf{P}}, \theta, D)$ and change the semantic loss ratio ρ_{sem} accordingly to operate over the formula forms of $\overline{\mathsf{P}}$ in Eq 6. By analogy to the generalized DPA in Eq 1, we can view this logistic log form as a particular instance of a **generalized semantic loss**:

Name	$f(\rho_{\text{sem}}, \beta) =$	**Semantic loss ratio** ρ_{sem}
$\ell_{\text{sl-log}}$	$-\log\sigma(\beta\rho_{\text{sem}})$	$\log \frac{\text{WMC}(\overline{\mathsf{P}_f};\theta)}{\text{WMC}(\overline{\neg\mathsf{P}}_f;\theta)}$
$\ell_{\text{sl-squared}}$	$(\rho_{\text{sem}} - \frac{1}{2\beta})^2$	
$\ell_{\text{sl-margin}}$	$\max(0, \beta - \rho_{\text{sem}})$	

Table 3: Different forms of the semantic loss that match the DPA losses in Table 1.

$$\ell_{\text{sl}}(\overline{\mathsf{P}}, \theta, D) := \mathop{\mathbb{E}}_{d\sim D}\Big[f(\rho_{\text{sem}}(d), \beta)\Big] \tag{7}$$

where, like in DPA, different choices can be made about what f to apply over the semantic loss ratio ρ_{sem}, which gives rise to several novel logics. To match the structure of DPA, we also add a weight parameter β. We define three particular versions of SL in Table 5, which we will need to apply our formal analysis to particular DPA losses in Table 1.

How many loss functions are there? Under this new formulation, we can view loss creation as a generative procedure, where we first select a f then sample two formulas P_1 and P_2 (each denoting a unique Boolean function in n variables) to create a $\overline{\mathsf{P}}$ via Prop 1. This view allows us to estimate the total number of definable loss functions to be doubly exponential in the number of probabilistic predictions n, equal to 4^{2^n} (i.e., the unique pairs of Boolean functions). For `DPO`, which involves four probabilistic predictions, this results in more than 4.2 billion variations that can be defined (how exactly losses like `DPO` are translated into preference structures is addressed in Section 5.2).

How is the loss space structured? While the space of loss functions is often very large, one can structure this space using the semantics of the corresponding formulas. Below we define preference entailment and equivalence and relate these semantic notions to the behavior of the compiled losses.

The following formal results (see proofs in Appendix B) give us tools for structuring the DPA loss space and informing the search for new loss functions.

We define **preference entailment** for two preference structures $\overline{\mathsf{P}}^1 \sqsubseteq \overline{\mathsf{P}}^2$ in terms of ordinary propositional entailment ($\models$) between formula forms: $\overline{\mathsf{P}}^1 \sqsubseteq \overline{\mathsf{P}}^2 := (\overline{\mathsf{P}_f}^1 \models \overline{\mathsf{P}_f}^2 \wedge \overline{\neg \mathsf{P}_f}^2 \models \overline{\neg \mathsf{P}_f}^1)$. Below we show (proof deferred to Appendix) that losses are monotonic w.r.t. preference entailment, as in the original SL (Xu et al., 2018).

Proposition 2 (monotonicity). *If* $\overline{\mathsf{P}}^{(1)} \sqsubseteq \overline{\mathsf{P}}^{(2)}$ *then* $\ell_{sl}(\overline{\mathsf{P}}^{(1)}, \theta, D) \geq \ell_{sl}(\overline{\mathsf{P}}^{(2)}, \theta, D)$ *for any* θ, D.

We will use later entailment to characterize the relative strength of DPA losses and visualize their relations using a representation called a **loss lattice** (see Figure 3). We also extend preference entailment to **preference equivalence** in a natural way: $\overline{\mathsf{P}}^1 \equiv \overline{\mathsf{P}}^{(2)} := (\overline{\mathsf{P}}^1 \sqsubseteq \overline{\mathsf{P}}^2 \wedge \overline{\mathsf{P}}^2 \sqsubseteq \overline{\mathsf{P}}^1)$. It follows as a corollary to the above proposition that our semantic loss is equivalent under prefrence equivalence, i.e., whenever $\overline{\mathsf{P}}^1 \equiv \overline{\mathsf{P}}^2$ then $\ell_{\text{sl}}(\overline{\mathsf{P}}^{(1)}, \theta, D) = \ell_{\text{sl}}(\overline{\mathsf{P}}^2, \theta, D)$ for any θ, D.

Finally, when comparing losses with differing numbers of probabilistic predictions or variables, we also prove a locality property that ensures that such a comparison is possible (see Prop 5).

5.2 Decompiling DPA losses into preference structures

The **decompilation** of a DPA loss ℓ_{DPA_x} into a symbolic form can now be stated as finding a preference structure $\overline{\mathsf{P}}$ whose semantic loss is equal to ℓ_{DPA_x}, as given in Eq 8:

$$\forall D, \theta.\ \ell_{\text{DPA}_x} = \ell_{\text{sl}}(\overline{\mathsf{P}}, D_{\text{p}}, \theta)\ \text{s.t.} \qquad (8) \qquad \rho_\theta = \rho_{\text{sem}},\ \text{with}\ \frac{\rho^t_\theta}{\rho^b_\theta} = \frac{\text{WMC}\left(\overline{\mathsf{P}}; \theta\right)}{\text{WMC}\left(\overline{\neg \mathsf{P}}; \theta\right)} \qquad (9)$$

We will say that a preference structure $\overline{\mathsf{P}}$ **correctly characterizes** a loss ℓ_x under some ℓ_{sl} whenever this condition holds. Given the structure of the DPA loss (Eq 1) and the generalized semantic loss (Eq 7), whenever f is fixed this can be reduced to finding a $\overline{\mathsf{P}}$ whose semantic loss ratio ρ_{sem} is equal to ℓ_x's core loss equation ρ_θ as shown in Eq 9.

Based on this, we define a procedure for translating the core loss equations ρ_θ in Table 2 into preference structures and ρ_{sem}. We consider each part in turn.

Algorithm 1: DPA to logic

Input : loss expression $\rho_\theta = \frac{\rho^t_\theta}{\rho^b_\theta}$

Output: $\overline{\mathsf{P}} = (\mathsf{P}, \mathsf{P}_{\mathbf{C}}, \mathsf{P}_{\mathbf{A}})$

1 $\mathsf{P}_t \leftarrow \text{SEM}(\rho^t_\theta)$
2 $\mathsf{P}_b \leftarrow \text{SEM}(\rho^b_\theta)$
3 $\mathsf{P} \leftarrow \text{SIMPLIFY}(\texttt{Implies}(\mathsf{P}_b, \mathsf{P}_t))$
4 $\mathsf{P}_{\mathbf{C}} \leftarrow \text{SIMPLIFY}(\texttt{Or}(\mathsf{P}_t, \mathsf{P}_b))$
5 $\mathsf{P}_{\mathbf{A}} \leftarrow \text{SIMPLIFY}(\texttt{And}(\mathsf{P}_t, \mathsf{P}_b))$

Characterizing the DPA equation class By construction, we will assume that all the core equations for DPA losses ρ^t_θ and ρ^b_θ are expressible as certain types of *disjoint* multilinear polynomials over binary variables $\{x_i\}_{i=1}^n$, intuitively polynomials whose translation via the rules in Table 7 results in valid formulas of propositional logic. Formally, such polynomials over n variables are defined as any polynomial e of the form $e = \sum_i e_i$ where (a) for all i there exists $J_i \subseteq \{1, \ldots, n\}$ such that $e_i = \prod_{j \in J_i} \ell_{ij}$ where ℓ_{ij} is either x_j or $(1 - x_j)$, and (b) for all i, i', terms e_i and $e_{i'}$ are disjoint, i.e., have no common solutions (for some k, one term has x_k and the other has $1 - x_k$).

We note that not all preference loss functions in the preference learning literature immediately fit this format, including the original form of DPOP (Pal et al., 2024) which we discuss in Appendix D and fix through **variable copying** as shown in Table 2.

Translation algorithm Our translation process is shown in Algorithm 1. Given an input ρ_θ, both parts of that equation are translated into logic (**lines 1-2**) via a translation function SEM. The translation is standard and its correctness can be established via induction on the rules (see the full rules in Table 7): each model prediction $P_{\texttt{M}}(\cdot)$ is mapped to a probabilistic prediction $\texttt{M}(\cdot)$ then: $1 - \mathsf{P}$ is mapped to negation, $\mathsf{P}_1 \cdot \mathsf{P}_2$ to conjunction, and $\mathsf{P}_1 + \mathsf{P}_2$ to disjunction. **Lines 3-5** apply the implication construction from Prop 1 to create a $\overline{\mathsf{P}}$, where formulas are minimized via SIMPLIFY.

The following result establishes the correctness of our decompilation procedure, which follows from the correctness of our translation rules and the implication construction from Prop 1.

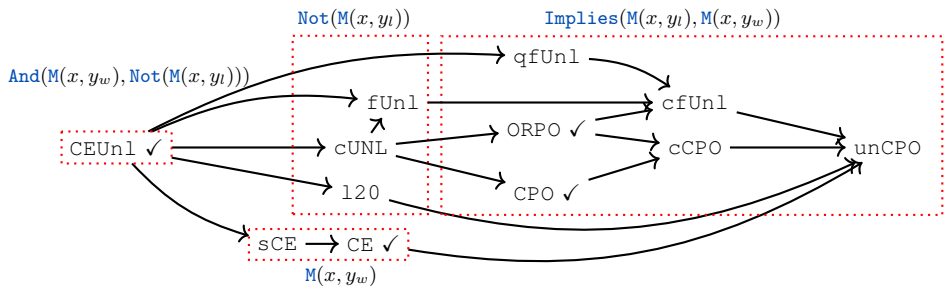

Figure 3: What other losses are there? Here we show the loss landscape for single model preference approaches using a **loss lattice** showing losses (nodes) structured according to strict entailment and their core formulas P (dashed boxes). See Appendix C for details of the individual losses.

Proposition 3 (correctness). *Given a loss equation $\rho_\theta = \rho^t_\theta / \rho^b_\theta$ where ρ^t_θ, and ρ^b_θ are disjoint multilinear polynomials, Algorithm 1 returns a preference structure $\overline{\mathsf{P}}$ whose semantic loss ratio ρ_{sem} equals ρ_θ.*

6 Results and Discussion

Table 4 shows the preference structures obtained from Algorithm 1 for the key DPA losses in Table 2. Given that the original losses were all formulated using the logistic log form of DPA, the correctness of Algorithm 1 (Prop. 3) tells us that compiling the representations in Table 4 under $\ell_{\text{sl-log}}$ will yield exactly the original losses. Importantly, when the DPO symbolic form is compiled using $\ell_{\texttt{sl-square}}$ (i.e., the squared loss form of SL), this will yield exactly IPO Azar et al. (2023), showing how our semantic analysis is invariant to the particular choice of f. (A similar argument can be made for deriving SliC from $\ell_{\text{sl-margin}}$ and the representation for CPO).

6.1 What we learn about existing losses?

Single model approaches have an intuitive semantics, highly constrained Under our analysis, CPO and ORPO are both derived from the same core semantic formula P and implication first introduced in Figure 2, in spite of the superficial differences in their original form. They differ, however, in terms of the conditioning constraints $\mathsf{P_C}$ they impose, with CPO imposing a **one-true** constraint that requires either the winner or loser to be deemed valid, whereas ORPO imposes a **one-hot** constraint where one and only one can be deemed valid. When plotted in a broader loss landscape, as shown in Figure 3, we see that both are entailed by the CEUNL baseline, yet have a non-entailing relation to one another.

Loss	Representation $\overline{\mathsf{P}}$
CE	$\mathsf{P} :=$ **M**(x,y_w), $\mathsf{P_C} := \bot$
CEUnl	$\mathsf{P} :=$ **And**(**M**(x,y_w), **Not**(**M**(x,y_l))), $\mathsf{P_C} := \bot$
CPO	$\mathsf{P} :=$ **Implies**(**M**(x,y_l), **M**(x,y_w)) $\mathsf{P_C} :=$ **Or**(**M**(x,y_l), **M**(x,y_w))
ORPO	$\mathsf{P} :=$ **Implies**(**M**(x,y_l),**M**(x,y_w)) $\mathsf{P_C} :=$ **Or**(**And**(**M**(x,y_l), **Not**(**M**(x,y_w))), **And**(**Not**(**M**(x,y_l)), **M**(x,y_w)))
DPO	$\mathsf{P} :=$ **Implies**(**And**(**Ref**(x,y_w),**M**(x,y_l)), **And**(**Ref**(x,y_l),**M**(x,y_w))) $\mathsf{P_C} :=$ **Or**(**And**(**Ref**(x,y_w),**M**(x,y_l)), **And**(**Ref**(x,y_l),**M**(x,y_w)))

Table 4: Formalizations of some of the losses from Table 2 shown in terms of P and $\mathsf{P_C}$ (for succinctness, we exclude $\mathsf{P_A}$ which can be inferred from each $\mathsf{P_C}$ via Algorithm 5.1).

In general, we see that all preference losses are highly constrained, which might explain their success. This is in sharp to contrast to the kinds of losses typically used with the semantic loss and neuro-symbolic modeling. For this reason, we think there is much to learned by working backward from empirically successful loss functions to their semantic properties to try and find out what properties make them successful and how they differ from conventional neuro-symbolic techniques.

There are many losses still to explore We systematically create new losses by manipulating the the conditioning constraints that existing losses impose. Figure 3 shows a (non-exhaustive) lattice representation of the loss landscape for single model preference approaches created by mechanically deriving new losses from the CEUnl baseline (i.e., the most constrained loss) and ordering them by strict entailment (this terminates in unCPO, a version of CPO without conditioning; see Appendix C for details). Here we see that different regions emerge characterized by different formulas P, notably an entirely unexplored region between between CEUnl and CPO and ORPO of unlikelihood losses that optimize for the negation of the loser ($\texttt{Not}(\texttt{M}(\mathrm{x}, \mathrm{y}_l))$).

DPO has a peculiar semantics, shared among variants The semantics of DPO shown in Table 4 is logically equivalent to a conjunction of two implications: $\texttt{Ref}(\mathrm{x}, \mathrm{y}_w) \wedge \texttt{M}(\mathrm{x}, \mathrm{y}_l) \rightarrow \texttt{M}(\mathrm{x}, \mathrm{y}_w)$ and $\texttt{Ref}(\mathrm{x}, \mathrm{y}_w) \wedge \neg\texttt{M}(\mathrm{x}, \mathrm{y}_l) \rightarrow \neg\texttt{M}(\mathrm{x}, \mathrm{y}_l)$. The first says that *If the reference deems the winner to be valid and the tunable model deems the loser to be valid, then that model should also deem the winner to be valid*, while the second says that *the tunable model should deem the loser to be not valid whenever the reference deems the winner to be valid and the loser to be not valid*. While this semantics makes sense, and complements nicely the semantics of CPO by adding information about the referent model, DPO includes conditioning constraints that are hard to justify from first principles, and that make it semantically disconnected from the CE and CEUnl baselines. While DPO belongs to a much larger space, we conjecture that investigating the different semantic neighborhoods that result from modifying its conditioning constraints, as in Figure 3, is a promising direction.

We also note that variants like SimPO and DPOP when formalized maintain exactly the same structure of DPO in Table 4, with DPOP adding repeated variables that amplify the score of the winner. Giving the semantic similarity between these variants and DPO, any small semantic change found in one would likely be useful in these others, which motivates general exploration into varying the conditioning constraints.

6.2 Are any of these new losses good?

The ultimate goal of our analysis is to facilitate the discovery of empirically improved versions of existing DPA losses. We hypothesize that the degree of constrainedness of a loss function, which is a natural property to characterize in our framework, is a key property underlying its success. This hypothesis is based on our initial empirical investigations into the new losses introduced in Figure 3, which we plan to explore further.

7 Conclusion

Despite the routine use of a variety of DPA algorithms to align LLMs with human preferences, knowing what exactly these the losses underlying these algorithms capture and how they relate to each other remains largely unknown. We presented a new technique for characterizing the semantics of such losses in terms of logical formulas over boolean propositions that capture model predictions. Key to our approach is the *decompilation* procedure, allowing one to derive provably correct symbolic formulas corresponding to any loss function expressed as a ratio of disjoint multilinear polynomials. Our approach provides a fresh perspective into preference losses, identifying a rich loss landscape and opening up new ways for practitioners to explore new losses by systematically varying the symbolic formulas corresponding to existing successful loss functions.

References

Kareem Ahmed, Stefano Teso, Kai-Wei Chang, Guy Van den Broeck, and Antonio Vergari. Semantic probabilistic layers for neuro-symbolic learning. *Advances in Neural Information Processing Systems*, 35:29944–29959, 2022.

Kareem Ahmed, Stefano Teso, Paolo Morettin, Luca Di Liello, Pierfrancesco Ardino, Jacopo Gobbi, Yitao Liang, Eric Wang, Kai-Wei Chang, Andrea Passerini, et al. Semantic loss functions for neuro-symbolic structured prediction. In *Compendium of Neurosymbolic Artificial Intelligence*, pp. 485–505. IOS Press, 2023.

Afra Amini, Tim Vieira, and Ryan Cotterell. Direct preference optimization with an offset. *arXiv preprint arXiv:2402.10571*, 2024.

Akari Asai and Hannaneh Hajishirzi. Logic-guided data augmentation and regularization for consistent question answering. *ACL 2020*, 2020.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*, 2023.

Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. Prompting is programming: A query language for large language models. *Proceedings of the ACM on Programming Languages*, 7(PLDI): 1946–1969, 2023.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172(6-7):772–799, 2008.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Dragan Cvetković. *The Logic Preference and Decision Supporting Systems*. PhD thesis, University of Beograd, 1993.

Luc De Raedt and Angelika Kimmig. Probabilistic (logic) programming concepts. *Machine Learning*, 100:5–47, 2015.

Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *IJCAI 2007, Proceedings of the 20th international joint conference on artificial intelligence*, pp. 2462–2467. IJCAI-INT JOINT CONF ARTIF INTELL, 2007.

David Dohan, Winnie Xu, Aitor Lewkowycz, Jacob Austin, David Bieber, Raphael Gontijo Lopes, Yuhuai Wu, Henryk Michalewski, Rif A Saurous, Jascha Sohl-Dickstein, et al. Language model cascades. *arXiv preprint arXiv:2207.10342*, 2022.

Jason Eisner, Eric Goldlust, and Noah A Smith. Dyna: A declarative language for implementing dynamic programs. In *Proc. of ACL*, 2004.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Daan Fierens, Guy Van den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt. Inference and learning in probabilistic logic programs using weighted boolean formulas. *Theory and Practice of Logic Programming*, 15(3):358–401, 2015.

Marc Fischer, Mislav Balunovic, Dana Drachsler-Cohen, Timon Gehr, Ce Zhang, and Martin Vechev. Dl2: training and querying neural networks with logic. In *International Conference on Machine Learning*, pp. 1931–1941. PMLR, 2019.

Dan Friedman, Alexander Wettig, and Danqi Chen. Learning transformer programs. *Advances in Neural Information Processing Systems*, 36, 2024.

Francesco Giannini, Giuseppe Marra, Michelangelo Diligenti, Marco Maggini, and Marco Gori. On the relation between loss functions and t-norms. In *Inductive Logic Programming: 29th International Conference, ILP 2019, Plovdiv, Bulgaria, September 3–5, 2019, Proceedings 29*, pp. 36–45. Springer, 2020.

Joseph Y Halpern, Robert Harper, Neil Immerman, Phokion G Kolaitis, Moshe Y Vardi, and Victor Vianu. On the unusual effectiveness of logic in computer science. *Bulletin of Symbolic Logic*, 7 (2):213–236, 2001.

John Hewitt, Michael Hahn, Surya Ganguli, Percy Liang, and Christopher D. Manning. RNNs can generate bounded hierarchical languages with optimal memory. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1978–2010, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.156.

Jiwoo Hong, Noah Lee, and James Thorne. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*, 2024.

Xiangkun Hu, Tong He, and David Wipf. New desiderata for direct preference optimization. *arXiv preprint arXiv:2407.09072*, 2024.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. A Logic-Driven Framework for Consistency of Neural Models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3924–3935, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10/ghvf88.

Ziyang Li, Jiani Huang, and Mayur Naik. Scallop: A language for neurosymbolic programming. *Proceedings of the ACM on Programming Languages*, 7(PLDI):1463–1487, 2023.

Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. *Advances in neural information processing systems*, 31, 2018.

Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, and Marco Gori. Integrating learning and reasoning with deep logic models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 517–532. Springer, 2019.

John McCarthy et al. *Programs with common sense*. RLE and MIT computation center Cambridge, MA, USA, 1960.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.

Nils J Nilsson. Logic and artificial intelligence. *Artificial intelligence*, 47(1-3):31–56, 1991.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Proceedings of Neurips*, 2024.

Nicholas Rescher. *The logic of decision and action*. University of Pittsburgh Press, 1967.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*, 2024.

Emile van Krieken, Samy Badreddine, Robin Manhaeve, and Eleonora Giunchiglia. Uller: A unified language for learning and reasoning. In *International Conference on Neural-Symbolic Learning and Reasoning*, pp. 219–239. Springer, 2024a.

Emile van Krieken, Pasquale Minervini, Edoardo M Ponti, and Antonio Vergari. On the independence assumption in neurosymbolic learning. *arXiv preprint arXiv:2404.08458*, 2024b.

Tim Vieira, Matthew Francis-Landau, Nathaniel Wesley Filardo, Farzad Khorasani, and Jason Eisner. Dyna: Toward a self-optimizing declarative language for machine learning applications. In *Proceedings of the 1st ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pp. 8–17, 2017.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*, 2023.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*, 2019.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*, 2024.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. A Semantic Loss Function for Deep Learning with Symbolic Knowledge. In *International Conference on Machine Learning*, pp. 5498–5507, 2018.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023.

Yao Zhao, Mikhail Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu. Calibrating sequence likelihood improves conditional language generation. In *The eleventh international conference on learning representations*, 2022.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. SLiC-HF: Sequence Likelihood Calibration with Human Feedback. *arXiv preprint arXiv:2305.10425*, 2023.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Input	$\text{SEM}(\cdot)$
predictions	
$\mathsf{P}_{\mathtt{M}}(\mathrm{y} \mid \mathrm{x})$	$\mathsf{P} := \mathtt{M}(\mathrm{x}, \mathrm{y})$
formulas P	
$\mathsf{P}_1 \cdot \mathsf{P}_2$	$\mathsf{P} := \mathtt{And}(\mathsf{P}_1, \mathsf{P}_2)$
$1 - \mathsf{P}$	$\mathsf{P} := \mathtt{Not}(\mathsf{P})$
$\mathsf{P}_1 + \mathsf{P}_2$	$\mathsf{P} := \mathtt{Or}(\mathsf{P}_1, \mathsf{P}_2)$

Table 5: Rules for the translation of loss expressions into symbolic formulas.

A SEMANTIC TRANSLATION RULES

In Table 7 we show the full translation rules for Algorithm 1.

B PROOFS OF PROPOSITIONS

Below we state propositions in Section 5.1 with their proofs.

Proposition 4 (monotonicity). *If* $\overline{\mathsf{P}}^{(1)} \sqsubseteq \overline{\mathsf{P}}^{(2)}$ *then* $\ell_{sl}(\overline{\mathsf{P}}^{(1)}, \theta, D) \geq \ell_{sl}(\overline{\mathsf{P}}^{(2)}, \theta, D)$ *for any* θ, D.

Proof. By the definition of preference entailment, we have $\overline{\mathsf{P}}_f^{(1)} \models \overline{\mathsf{P}}_f^{(2)}$. This means that for any d, $\overline{\mathsf{P}}^1(d) \models \overline{\mathsf{P}}^2(d)$, which implies that for any θ, $\text{WMC}\big(\overline{\mathsf{P}}^{(1)}(d); \theta\big) \leq \text{WMC}\big(\overline{\mathsf{P}}^{(2)}(d); \theta\big)$. From the definition of preference entailment, we also have $\overline{\neg\mathsf{P}}^{(2)}(d) \models \overline{\neg\mathsf{P}}^{(1)}(d)$. Following a similar line of reasoning as above, this implies $\text{WMC}\big(\overline{\neg\mathsf{P}}^{(1)}(d); \theta\big) \geq \text{WMC}\big(\overline{\neg\mathsf{P}}^{(2)}(d); \theta\big)$. Thus, for any d and θ, the weighted model counting ratio term in the semantic loss in Table 5 is no larger for $\overline{\mathsf{P}}^{(1)}$ than for $\overline{\mathsf{P}}^{(2)}$. It follows that $\ell_{\text{sl}}(\overline{\mathsf{P}}^{(1)}, \theta, \{d\}) \geq \ell_{\text{sl}}(\overline{\mathsf{P}}^{(2)}, \theta, \{d\})$. Taking the expectation over $d \sim D$, we obtain $\ell_{\text{sl}}(\overline{\mathsf{P}}^{(1)}, \theta, D) \geq \ell_{\text{sl}}(\overline{\mathsf{P}}^{(2)}, \theta, D)$. □

Proposition 5 (locality). *Let* $\overline{\mathsf{P}}$ *be a preference structure defined over probabilistic prediction variables* $\mathbf{X}$ *with parameters* θ_x. *Let* $\mathbf{Y}$ *be some disjoint set of variables with parameters* θ_y. *Then* $\ell_{sl}(\overline{\mathsf{P}}, \theta_x, D) = \ell_{sl}(\overline{\mathsf{P}}, [\theta_x\, \theta_y], D)$ *for any* D.

Proof. Let $\mathbf{w}_x$ be any world over variables $\mathbf{X}$ and $\mathbf{w}_y$ be any world over (disjoint) variables $\mathbf{Y}$. Let $\mathbf{w}_{x,y}$ denote the joint world. By Eq 2, the probability of the world $\mathbf{w}_{x,y}$ in the $(\mathbf{X}, \mathbf{Y})$ space can be written as $\mathrm{P}_{\theta_x,\theta_y}(\mathbf{w}_{x,y}) = \prod_{X_i \in \mathbf{X}} Q_{\theta_x,\theta_y}(X_i) \cdot \prod_{Y_j \in \mathbf{Y}} Q_{\theta_x,\theta_y}(Y_j)$ where Q is either P or $1 - \mathrm{P}$. Since the parameters θ_x and θ_y refer to disjoint sets of variables, we can simplify this to $\prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) \cdot \prod_{Y_j \in \mathbf{Y}} Q_{\theta_y}(Y_j)$.

It follows that the marginal probability of the world $\mathbf{w}_x$ in the $(\mathbf{X}, \mathbf{Y})$ space equals $\mathrm{P}_{\theta_x,\theta_y}(\mathbf{w}_x) = \sum_{\mathbf{Y}} \left(\prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) \cdot \prod_{Y_j \in \mathbf{Y}} Q_{\theta_y}(Y_j)\right) = \prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) \cdot \sum_{\mathbf{Y}} \left(\prod_{Y_j \in \mathbf{Y}} Q_{\theta_y}(Y_j)\right) = \prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) \cdot \prod_{Y_j \in \mathbf{Y}} \left(Q_{\theta_y}(Y_j) + (1 - Q_{\theta_y}(Y_j))\right) = \prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) = \mathrm{P}_{\theta_x}(\mathbf{w}_x)$. This last expression is precisely the probability of the world $\mathbf{w}_x$ in only the $\mathbf{X}$ space. Thus, $\mathrm{P}_{\theta_x}(\mathbf{w}_x) = \mathrm{P}_{\theta_x,\theta_y}(\mathbf{w}_x)$, which implies $\text{WMC}\big(\overline{\mathsf{P}}; \theta_x\big) = \text{WMC}\big(\overline{\mathsf{P}}; \theta_x, \theta_y\big)$ and similarly for $\overline{\neg\mathsf{P}}$. From this, the claim follow immediately. □

C BOOLEAN VISUALIZATION OF LOSSES

To visualize the semantics of the single model losses shown in Figure 3, we can use a Boolean truth table representation as shown in Figure 4. Here each column shows a specific loss function representable as a preference structure $\overline{\mathsf{P}}$. Intuitively, ✓ shows all the propositional models to count that are connected with the formula form of $\overline{\mathsf{P}}$ (or are in the numerator of the semantic loss ratio) and

✗ shows all the propositional models to count that are connected with the negated formula form (or the denominator in the semantic loss ratio).

Putting this together, we can loosely define the logistic form of the semantic loss as follows (where WCOUNT refers to the weight count of rows either with ✓ or ✗):

$$-\log\sigma\left(\log\frac{\text{WCOUNT}(\checkmark)}{\text{WCOUNT}(\times)}\right)$$

D DPOP EQUATION

The DPOP loss function in Table 2 adds to the DPO an additional log term $\alpha \cdot \max(0, \log \frac{\text{P}_{\text{ref}}(y_w|x)}{\text{P}_\theta(y_w|x)})$ that aims to ensure that the log-likelihood of preferred example is high relative to the reference model (we simplified this loss by removing the $\max$ and α parameter, the latter of which is set to be a whole number ranging from 5 to 500 in Pal et al. (2024)). When translating the full loss into a single $\log$, this results in the equation $\rho_\theta = \log \frac{\text{P}_{\text{ref}}(y_l|x)\text{P}_\theta(y_w|x)^2}{\text{P}_{\text{ref}}(y_w|x)^2\text{P}_\theta(y_l|x)}$ for $\alpha = 1$. The top and bottom equations are hence not multilinear since they both contain exponents > 1. To fix this, we can simply create copies of these variables, e.g., with $\text{P}_\theta\left(\text{x}, \text{y}_\text{w}\right)^2$ and $\text{P}_{\text{ref}}(y_l \mid x)^2$ set to $\text{P}_\theta\left(\text{x}, \text{y}_\text{w}\right)\text{P}_{\theta 2}\left(\text{x}, \text{y}_\text{w}\right)$ and $\text{P}_{\text{ref}}(y_l \mid x)\text{P}_{\text{ref2}}(y_l \mid x)$ using the copied prediction variables $\text{P}_{\theta 2}(\cdot)$ and $\text{P}_{\text{ref2}}(\cdot)$.

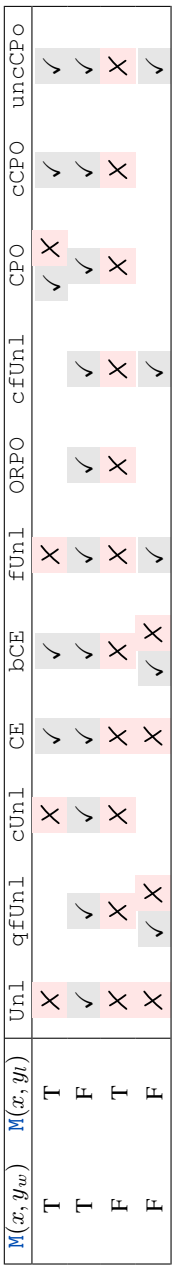

Figure 4: A Boolean representation of the different losses covered in Figure 3