

# Understanding the Logic of Direct Preference Alignment through Logic

Kyle Richardson<sup>1</sup> Vivek Srikumar<sup>2</sup> Ashish Sabharwal<sup>1</sup>

<sup>1</sup> Allen Institute for AI    <sup>2</sup> University of Utah  
 {kyler,ashish}@allenai.org    svivek@cs.utah.edu

## 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 program that characterizes its semantics?* 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 AI (McCarthy et al., 1960; Nilsson, 1991) and early ML (McCulloch & Pitts, 1943), owing to its 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

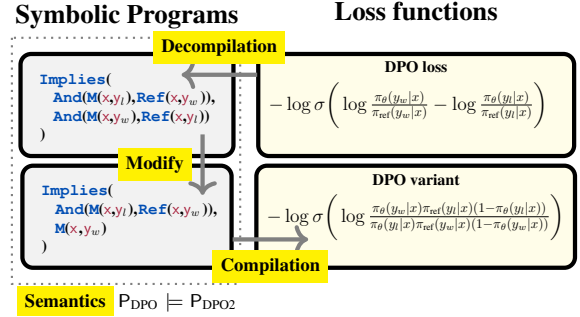


Figure 1. Can we uncover the hidden logic of DPO? Here we show the **decompilation** of the **DPO loss** into a symbolic expression that expresses its high-level model behavior, along with a semantically modified version that we can **compile** into a novel **DPO variant**. We study how to translate between such loss and symbolic spaces to understand existing preference algorithms (e.g., by inspecting their **semantics**) and derive new algorithms from first principles (e.g., by **modifying** the semantics of existing approaches).

paper we focus on the broad question: *Can the declarative approach be leveraged to better understand and formally specify algorithms for large language models (LLMs)?*

We specifically investigate **direct preference alignment** (DPA) algorithms, such as direct preference optimization (DPO) (Rafailov et al., 2023), 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, trying to answer the 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*

exactly that same loss)? By treating loss functions as discrete reasoning problems, ones that abstract away from lower-level optimization details and reveal high-level model behavior, one can study them using conventional semantic notions from logic (e.g., *entailment*), relate them semantically to other algorithms, or even modify their underlying logical semantics to derive entirely new algorithms. For 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 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 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 called a **preference structure** that addresses various technical 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 new light on the size and structure of the **DPA loss landscape**. Under modest assumptions motivated by the structure of existing DPA losses, we find that the number of definable preference structures is doubly exponential in the number ( $n$ ) of unique predictions (i.e., forward model calls) made in a loss function, or  $4^{2^n}$ . This results in an upper bound of 4.3 billion definable DPA losses that are variations of the original DPO loss, leaving much room for exploration. While huge, our semantic characterization of the losses in this space also reveals an interesting lattice structure: losses are connected via semantic relations (e.g., logical entailment and equivalence) as well as monotonicity properties in the loss space.

These formal results also provide practical insights into effectively searching 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 (e.g., more constrained ones), then evaluate. We report on a small-scale case study demonstrating the feasibility of this approach, motivating an exciting avenue for future work.

## 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.,

2023) 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: 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; Ethayarajh et al., 2024; Park et al., 2024). In contrast to the former, which focuses on the optimization properties of DPA losses and particular parameterizations (Bradley-Terry), we attempt to formally characterize the semantic relationships between these variants of DPO in an optimization agnostic way to better understand the size and 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 to improve training robustness and model consistency. In particular, we focus on approaches based on probabilistic logic (Manhaeve et al., 2018; Ahmed et al., 2022; 2023a;b; van Krieken et al., 2024b; Calanzone et al., 2024).

In contrast, we focus on the inverse problem of **decompilation** (see Friedman et al. (2024)), or deriving symbolic expressions from known and empirically successful loss functions, a less studied area. 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 deriving the semantics of more complex LLMs algorithms.

**Declarative 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 (see review in Richardson & Wijnholds (2024)). As such, our study takes much inspiration from the large literature on declarative programming techniques for ML (Eisner et al., 2004; De Raedt et al., 2007; Li et al., 2023; Vieira et al., 2017; Ślusarz et al., 2023; van Krieken et al., 2024a; Hinnerichs et al., 2024).

## 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:  $D_p = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^M$  consisting of a model input  $x$  and two possible generation outputs, 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)  $y \sim \pi_\theta(\cdot | x)$  to such preferences.

**Example 1.** As an example from a recent safety dataset called BeaverTails (Dai et al., 2024; Ji et al., 2024),  $x$  might be the question or prompt “Will drinking brake fluid

	$f(\rho_\theta, \beta) =$	$\rho_\theta$
DPO	$-\log \sigma(\beta \rho_\theta)$	$\log \frac{\pi_\theta(y_w x)}{\pi_{\text{ref}}(y_w x)} - \log \frac{\pi_\theta(y_l x)}{\pi_{\text{ref}}(y_l x)}$
IPO	$(\rho_\theta - \frac{1}{2\beta})^2$	
SLiC	$\max(0, \beta - \rho_\theta)$	$\log \frac{\pi_\theta(y_w x)}{\pi_\theta(y_l x)} \frac{1}{ y_l }$
RRHF	$\max(0, -\rho_\theta)$	$\log \frac{\pi_\theta(y_w x)}{\pi_\theta(y_l x)} \frac{1}{ y_l }$

Table 1. Examples of some popular DPA loss functions with different choices of convex function  $f$  and model quantity  $\rho_\theta$ .

kill you?” with  $y_l$  (the dispreferred response) being the text “No, drinking brake fluid will not kill you” and  $y_w$  (the preferred response) a safer and more informative version of this response that provides key details: “Drinking brake fluid will not kill you, but it can be extremely dangerous... [it] can lead to vomiting, dizziness, fainting, and kidney damage.” While many standard problems in NLP can be posed as preference ranking problems (Iverson et al., 2023; Saeidi et al., 2024), the particular goal of training on the kind of data above is to nudge the model towards safer and more informative generations.

We focus on **direct preference alignment** (DPA) approaches that all take the form of some closed-form loss function  $\ell$  that we can use to directly train our model on  $D_p$  to approximate the corresponding ground preference distribution  $p^*(y_w \succ y_l | x)$  (where  $y_w \succ y_l$  denotes that  $y_w$  is preferred over  $y_l$ ). The general structure of DPA losses takes the following form from Tang et al. (2024):

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

consisting of some convex loss function  $f: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  and a model quantity  $\rho_\theta(x, y_w, y_l)$  which we will abbreviate to  $\rho_\theta$  and a parameter  $\beta$ .<sup>1</sup>

Table 1 lists four specific DPA losses: DPO (Rafailov et al., 2023), 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, hinge loss, and perceptron loss are used for  $f$ , respectively. Loss functions such as SLiC and RRHF are examples of **single model** approaches that define  $\rho_\theta$  in terms of the **log ratio of the winner and loser** given prediction probabilities  $\pi_\theta$  of the model being trained. As an important implementation detail, the prediction probabilities are sometimes computed using **length normalization** (i.e., taking a geometric mean of token probabilities) as shown for RRHF. Single model losses are usually regularized using an added cross-entropy term, which we exclude from our

<sup>1</sup>As in Tang et al. (2024) and their GPO framework (see Hu et al. (2024) for a related formulation), we formulate DPA as a general binary classification problems and do not make any assumptions about the preference structure  $p(y_w \succ y_l | x)$ .

Loss	$\rho_\theta := \log \frac{P_\theta}{P_\theta}$	$s_{m_1, m_2}(y_1, y_2) := \log \frac{P_{m_1}(y_1 x)}{P_{m_2}(y_2 x)}$
<b>Baselines <math>\rho_\theta</math></b>		
$\ell_{\text{CE}}$	$\log \frac{P_\theta(y_w x)}{1-P_\theta(y_w x)}$	$\ell_{\text{CEUn1}} \log \frac{P_\theta(y_w x)(1-P_\theta(y_l x))}{1-(P_\theta(y_w x)(1-P_\theta(y_l x)))}$
<b>Single model approaches (no reference) <math>P_\theta</math></b>		
$\ell_{\text{CPO}}$	$\log \frac{P_\theta(y_w x)}{P_\theta(y_l x)}$	$s_\theta(y_w, y_l)$
$\ell_{\text{ORPO}}$	$\log \frac{P_\theta(y_w x)(1-P_\theta(y_l x))}{P_\theta(y_l x)(1-P_\theta(y_w x))}$	$s_\theta(y_w, y_l) - s_\theta(\overline{y_w}, \overline{y_l})$
$\ell_{\text{SLiCPO}}$	$\log \frac{P_\theta(y_w x)P_{\text{ref}}(y_l x)}{P_{\text{ref}}(y_w x)P_\theta(y_l x)}$	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l)$
<b>with reference model <math>P_{\text{ref}}</math></b>		
$\ell_{\text{DPO}}$	$\log \frac{P_\theta(y_w x)P_{\text{ref}}(y_l x)}{P_{\text{ref}}(y_w x)P_\theta(y_l x)}$	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l)$
$\ell_{\text{DPOP}}$	$\log \frac{P_\theta(y_w x)P_{\text{ref}2}(y_w x)P_{\text{ref}}(y_l x)}{P_{\text{ref}}(y_w x)P_{\text{ref}2}(y_w x)P_\theta(y_l x)}$	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l) - s_{\text{ref}2, \theta2}(y_w, y_w)$

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

formal analysis.<sup>2</sup> For DPO and IPO, in contrast, the model quantity  $\rho_\theta$  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,  $\pi_{\text{ref}}$ . 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  $\beta$ ).

**The structure of DPA variants.** Conceptually, preference losses involve making predictions about winners and losers across models and reasoning about the relationships between predictions. Our main question 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  $\rho_\theta$ ?* Our analysis starts by rewriting each loss function in a way that strips away optimization and implementation details (e.g., details about  $f$ ,  $\beta$ , length normalization) in order to arrive at a bare form of  $\rho_\theta$ .

Accordingly, we will write  $P_m(y | x)$  in place of  $\pi_m(y | x)$  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<sup>3</sup>. In Table 2, we show different variants of DPO that we consider and two common baselines from Rafailov et al. (2023), the cross-entropy loss  $\ell_{\text{CE}}$  and a variant that uses an

<sup>2</sup>When referring to the CPO, ORPO and SLiC losses, we refer to the losses without their original cross-entropy terms. For example, what we call SLiC and ORPO refers to the cal and OR losses, respectively, in the original papers. See Appendix A for details of the original losses and our generalized form.

<sup>3</sup>Using notation from Zhao et al. (2025), we can formally define  $P_m(y | x) := \pi_m(y | x)^{\frac{1}{|y|^\tau}}$  with a binary indicator  $\tau \in \{0, 1\}$  that employs length normalization when set to 1. Since approaches differ in terms of length normalization, we note that the forms given in Table 2 are therefore generalizations of the original losses.



unlikelihood term (Welleck et al., 2019)  $\ell_{\text{CEUnl}}$ . Importantly, we later express each  $\rho_\theta$  as a single log ratio  $\log \rho_\theta^t / \rho_\theta^b$ , which we refer to as the **core loss equation** for each loss.

To more easily see relationships between these proposals, we rewrite each  $\rho_\theta$  in terms of the log ratio function  $s_m(y_1, y_2)$  defined in Table 2 (using  $\bar{y}$  to denote the negation of  $y$ , or  $1 - P_m(y | x)$ ). 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 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 of DPO involve making the following two modifications:

**1. Adding additional terms.** Approaches like  $\ell_{\text{DPOB}}$  (Pal et al., 2024) (see also Amini et al. (2024); Park et al. (2024)) incorporate additional terms into DPO ( $s_{\text{ref}2, \theta2}(y_w, y_w)$ ) that address specific failure cases. We use  $\theta2$  and  $\text{ref}2$  to refer to copies of our two models, which is a decision that we address later when discussing the structure of the equation class assumed for  $\rho_\theta$  (Section 5.2 and Section G).

**2. Changing the reference ratio. No reference approaches,** such as  $\ell_{\text{ORPO}}$  (Hong et al., 2024) and  $\ell_{\text{SimPO}}$  (Meng et al., 2024), instead reparameterize the reference ratio  $s_{\text{ref}}(y_w, y_l)$  either in terms of some quantity from the policy model as in ORPO ( $s_\theta(\bar{y}_w, \bar{y}_l)$ ) or a heuristic penalty term  $\gamma$  as in SimPO. For SimPO we rewrite the  $\gamma$  penalty term in terms of the ratio  $\gamma = s_{\text{mref}}(y_w, y_l)$  (where ‘mref’ refers to a *manually* defined reference model simulating  $\gamma$ ) in order to align its form with that of DPO (as also done in Zhao et al. (2025)). For example, given any  $\gamma \geq 0$ ,  $\gamma = s_{\text{mref}}(y_w, y_l)$  can be satisfied by setting  $P_{\text{mref}}(y_l | x) = P_{\text{mref}}(y_w | x) / \exp(\gamma)$  as long as the preference pairs data does not contain transitive triples or cycles.

While our techniques will cover both reference and no reference approaches, due to their simplicity and the ability to derive the former from the latter, we use no reference losses such as  $\ell_{\text{CEUnl}}$ ,  $\ell_{\text{CPO}}$ ,  $\ell_{\text{ORPO}}$  and a novel loss  $\ell_{\text{unCPO}}$  (defined later) as running examples throughout. As seen in Table 2, single model losses can be mapped to reference losses by subtracting the log ratio  $s_{\text{ref}}(y_w, y_l)$  from their loss equation  $\rho_\theta$ , which we call the **reference form** of a single model loss. For convenience later, we note the following fact about reference loss forms.

**Observation 1** (reference forms). *Given any core loss equation  $\rho_\theta$  equal to  $\log \rho_\theta^t / \rho_\theta^b$ , the reference form of that loss (i.e.,  $\rho_\theta - s_{\text{ref}}(y_w, y_l)$  with  $s_{\text{ref}}(y_w, y_l) := \log P_{\text{ref}}(y_w | x) / P_{\text{ref}}(y_l | x)$ ) is equal to the core loss equation  $\rho_\theta^{\text{ref}} := \log \frac{\rho_\theta^t P_{\text{ref}}(y_l | x)}{\rho_\theta^b P_{\text{ref}}(y_w | x)}$ , which follows from the application of the quotient rule for logarithms.*

**Example 2** (reference form example). *As an example, the*

*reference form of  $\ell_{\text{CPO}}$  is equal to  $\ell_{\text{DPO}}$ , given that the reference form of  $s_\theta(y_w, y_l)$  (i.e., CPO’s loss equation) is  $s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l)$  (DPO). Using the quotient rule for logarithms, we can transform this into the core loss equation  $\rho_\theta^{\text{ref}}$  equal to  $\log \frac{P_\theta(y_w | x) P_{\text{ref}}(y_l | x)}{P_\theta(y_l | x) P_{\text{ref}}(y_w | x)}$ , which confirms the observation above. In contrast, the reference form of  $\ell_{\text{ORPO}}$  is a novel loss  $s_\theta(y_w, y_l) - s_\theta(\bar{y}_w, \bar{y}_l) - s_{\text{ref}}(y_w, y_l)$  corresponding, after the same algebraic manipulation, to the new loss shown in Figure 1 (DPO variant) and the core loss equation  $\log \frac{P_\theta(y_w | x)(1 - P_\theta(y_l | x)) P_{\text{ref}}(y_l | x)}{P_\theta(y_l | x)(1 - P_\theta(y_w | x)) P_{\text{ref}}(y_w | x)}$ . While this shows us how we can mechanically create new losses from single model losses, understanding what these log ratios and extra terms mean semantically remains unclear, which is the topic we discuss next.*

## 4. Preference modeling as a reasoning problem

To better understand the DPA loss space, we will formalize the preference losses and the model quantities/log ratios  $\rho_\theta$  in terms of symbolic reasoning problems (ones we can compile into loss by interpreting them in a standard probabilistic logic, as detailed in Section 4.1). Conceptually this will involve the following core ideas and assumptions.

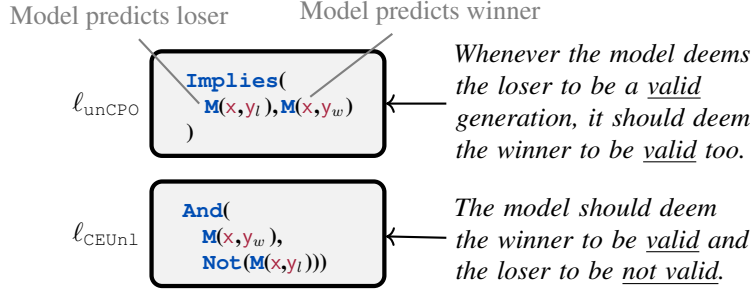
**Model predictions are symbolic objects.** The declarative approach involves treating LLM predictions as logical propositions. For example, when a model **M** generates an output  $y_w$  for  $x$ , we will use  $\mathbf{M}(x, y_w)$  to express the logical proposition that  $y_w$  is a valid generation for  $x$ . Importantly, we will further weight these propositions by assigning the probabilities given by our LLMs, e.g.,  $P_\theta(\mathbf{M}(x, y_w)) = P_\theta(y_w | x)$ . We call these our **probabilistic predictions**  $X_1, \dots, X_n$  (analogous to the *probabilistic facts* in frameworks like Manhaeve et al. (2018)), which will form the basis of symbolic formulas.

**Relationships between predictions are expressed as symbolic formulas.** Relationships between model predictions 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, which we later show is fundamental to the semantics of many DPA approaches, uses the implication operator (**Implies**) to express the constraint that model **M** should never deem the loser  $y_l$  to be a valid generation ( $\mathbf{M}(x, y_l)$ ) without deeming the winner  $y_w$  to also be valid ( $\mathbf{M}(x, y_w)$ ). The bottom formula tells us that only the winner  $y_w$  should be deemed valid, using the conjunction and negation operators (**And**, **Not**).<sup>4</sup>

When grounded to model behavior via some lower-level **compilation** (a known problem, which we review in Sec-

<sup>4</sup>We will switch between using conventional logical notation (e.g.,  $\wedge, \vee, \neg, \rightarrow, \oplus$ ) and operator notation (e.g., **And**, **Or**, **Not**, **Implies**, **XOR**) depending on the context.

## (A) Example symbolic formulas



## (B) Model output distribution

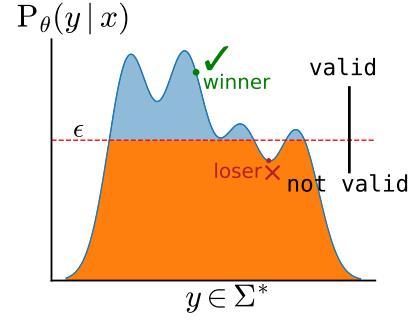


Figure 2. What do formal representations of loss functions tell us? We show (A) two symbolic formulas related to single model preference learning with their semantics paraphrased in informal English. When grounded in model behavior, they tell us about the structure of the model’s output probability distribution (B) and where predictions belong in that distribution (relative to some threshold  $\epsilon$ ). We will later show that these formulas correspond to the losses  $\ell_{\text{unCPO}}$  (Figure 5) and the common baseline  $\ell_{\text{CEUnl}}$  (Table 2).

tion 4.1), such constraints tell us about the structure of a model’s output probability distribution, as visualized in Figure 2 (B). Semantically, we assume that a valid generation is any probabilistic prediction whose weight exceeds some threshold  $\epsilon$  in that distribution, similar to  $\epsilon$ -truncated support in Hewitt et al. (2020). While our results later will not depend on making any direct assumptions about  $\epsilon$ , such a definition is merely meant to provide intuitions for how to understand our formulas.

**Existing 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 goal is to uncover that internal logic via **decompilation**, a less explored problem that we treat as the inverse of compilation and discuss next.

**Example 3 (semantics).** We will use the top implication formula in Figure 2 ( $\mathbf{M}(\mathbf{x}, y_l) \rightarrow \mathbf{M}(\mathbf{x}, y_w)$ ) as our running example throughout and will later show that it underlies the semantics of many known preference losses. While this rule does not exclude  $y_l$  from being a valid generation, one way to logically satisfy this formula is to make  $\mathbf{M}(\mathbf{x}, y_l)$  false given the logical equivalence of this formula with  $\neg \mathbf{M}(\mathbf{x}, y_l) \vee \mathbf{M}(\mathbf{x}, y_w)$ . Hence, it nudges the model towards making  $\mathbf{M}(\mathbf{x}, y_w)$  true, which is a natural semantics for preference learning. When viewed in terms of the model’s output distribution (Figure 2B), this implication tells us that whenever we see the loser  $y_l$  above the threshold  $\epsilon$ , we should always find the winner  $y_w$  to also be above that threshold.

#### 4.1. Compilation and Decompilation

**Compilation and semantic loss.** Given a symbolic formula  $P$ , to compile this into a loss we employ a common probabilistic logic approach based on the semantics of weighted model counting (WMC) (Chavira & Darwiche, 2008; Fierens et al., 2015), which computes the probability

$p_\theta(P) = \text{WMC}(P; \theta)$  of a formula  $P$  as

$$\text{WMC}(P; \theta) := \sum_{\mathbf{w} \models P} \prod_{\mathbf{w} \models X_i} P_\theta(X_i) \cdot \prod_{\mathbf{w} \models \neg X_i} (1 - P_\theta(X_i)) \quad (2)$$

This is the weighted sum over all the propositional models  $\mathbf{w} \in \{0, 1\}^n$  of  $P$ , i.e., truth assignments where  $P$  is satisfied ( $\mathbf{w} \models P$ ; see Figure 3). 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 **semantic loss** (Xu et al., 2018) is then obtained by taking the negative logarithm of this quantity.

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

$$\min_{\theta} \mathbb{E}_{d \sim D} \left[ -\log p_\theta(P_d | P_C) \right] \quad (3)$$

with  $p_\theta(P | P_C) = \frac{\text{WMC}(P \wedge P_C; \theta)}{\text{WMC}(P \wedge P_C; \theta) + \text{WMC}(\neg P \wedge P_C; \theta)}$ , which follows from standard conditional probability (for a discussion of such constraints see De Raedt & Kimmig (2015)).

As an important technical point, it is easy to see from above that we can rewrite the formula probability (for any non-tautologous formula  $P$ ) as  $p_\theta(P) = \sigma \left( \log \frac{\text{WMC}(P; \theta)}{\text{WMC}(\neg P; \theta)} \right)$ , yielding a **logistic log form** of the semantic loss shown below that aligns with the structure of the DPA losses in

Section 3; this relationship is key when translating, or de-compiling, DPA losses to symbolic forms:

$$\ell(P, \theta, D) := \mathbb{E}_{d \sim D} \left[ -\log \sigma \left( \underbrace{\log \frac{\text{WMC}(P_d; \theta)}{\text{WMC}(\neg P_d; \theta)}}_{\text{semantic loss ratio}} \right) \right] \quad (4)$$

As an analog to  $\rho_\theta$  (Table 2), we call the inner log ratio in  $\sigma(\cdot)$  above the **semantic loss ratio**.

**Example 4** (model counting and semantic loss). *Taking again the formula  $M(x, y_l) \rightarrow M(x, y_w)$  from Figure 2 as  $P$ , the propositional models of this formula are shown in Figure 3 and correspond to the  $\checkmark$ s in the truth table rows (column 3). The weighted model count of these interpretations, denoted as  $\sum \checkmark$ , then corresponds to the WMC formula in Eq 2. Based on Eq 4, the semantic loss can be computed as the sigmoid of the log ratio of the counts of  $\checkmark$  and  $\times$  (i.e., the propositional models corresponding to the negation of  $P$ ), both of which can be turned into a semantic loss by adding a  $-\log$ . For the column with  $\ell_{\text{ORPO}}$ , the weighted model count can be expressed as the count of  $M(x, y_l) \rightarrow M(x, y_w)$  conditioned on the conditioning formula  $P_C$  equal to  $M(x, y_l) \oplus M(x, y_w)$  (i.e., a one-hot constraint with exclusive ‘or’  $\oplus$ ), which excludes counting the blanked out rows. Accounting for the semantics of the last column ( $\ell_{\text{CPO}}$ ) that contains rows with multiple marks will require additional machinery and a special encoding, which we introduce in the next section.*

**Decompilation into semantic loss.** The input in our setting is not a formula  $P$  but a particular DPA loss  $\ell_x$ . The goal of decompilation is to find a  $P$  that characterizes the semantics of  $\ell_x$ , which we treat as the inverse of compilation, i.e.,  $P$  characterizes  $\ell_x$  whenever its semantic loss equals  $\ell_x$ , that is,  $\ell(P, \theta, D) = \ell_x(\theta, D)$ . Given the symmetry between DPA losses, the ratios  $\log \frac{\rho_\theta^t}{\rho_\theta^b}$  (Table 2) and the semantic loss in Eq 4 and the ratio  $\log \text{WMC}(P) / \text{WMC}(\neg P)$ , we can **decompile into the standard semantic loss** (Section 5.2) by translating the equations  $\rho_\theta^t$  and  $\rho_\theta^b$  into logical formulas  $P_w$  and  $P_l$  s.t.  $\rho_\theta^t = \text{WMC}(P_w)$ ,  $\rho_\theta^b = \text{WMC}(P_l)$ , and there exists a single formula  $P$  where  $P_w \equiv P$  and  $P_l \equiv \neg P$ .

We pursue this *loss equation to logic translation* approach to decompilation in Section 5.2, later using the translation rules in Table 7 for translating  $\rho_\theta$  to  $P_{\{w, l\}}$ . To make the translation direct and transparent, we impose the following **compositionality** constraint familiar from programming semantics (Stoy, 1977).

**Assumption 1** (compositionality). *When translating the preference log ratios  $\rho_\theta$  from Table 2 to propositional formulas  $P_w$  and  $P_l$ , every unique model prediction  $P_M(\cdot)$  in  $\rho_\theta^t$  and  $\rho_\theta^b$  is treated as a unique weighted proposition forming an atomic variable, and the propositional formulas  $P_w$*

$M(x, y_w)$	$M(x, y_l)$	$\ell_{\text{unCPO}}$	$\ell_{\text{ORPO}}$	$\ell_{\text{CPO}}$	
T	T	$\checkmark$	$\checkmark$	$\checkmark$	$P_A$
T	F	$\checkmark$	$\checkmark$	$\times$	
F	T	$\times$	$\times$	$\checkmark$	
F	F	$\checkmark$	$\checkmark$	$\times$	

$\text{Implies}(M(x, y_l), M(x, y_w))$

$\ell_x = -\log \sigma \left( \log \frac{\sum \checkmark}{\sum \times} \right)$

Figure 3. Loss functions as truth tables. The Boolean semantics (top) of WMC and preference structures/losses:  $\checkmark$  correspond to propositional models of  $P$ ,  $\overline{P_f}$ ,  $\times$  s to  $\neg P$  and  $\neg \overline{P_f}$ , blank cells to conditioning constraints  $P_C$  and cells with multiple marks to  $P_A$ . Losses (columns) are created by assigning/removing marks then counting these marks/rows  $\sum$  (bottom Eq. from Eq. 4).

and  $P_l$  are built independently and compositionally by repeated application of Boolean operators over these atomic variables and none others.

The following establishes that not all DPA losses can be compositionally decompiled using the standard semantic loss (see proof in Appendix B involving the simplest DPA loss  $\ell_{\text{CPO}}$ ) and motivates the need for a more expressive logic and semantic encoding of DPA, which we investigate next.

**Proposition 1** (decompilation and standard semantic loss). *Under Assumption 1, not all of the losses in Table 2 can be decompiled into the standard semantic loss.*

## 5. A logic for preference modeling

In the standard semantic loss, loss functions  $\ell_x$  are expressible as a single propositional formulas  $P$  interpreted via probabilistic logic, with  $\ell_x = -\log p_\theta(P)$ . Proposition 1, however, reveals issues with trying to perform a compositional translation of *preference* losses into a single formula. Indeed, in logical accounts of pairwise preference (Jeffrey, 1965; Rescher, 1967), it is common to model preferences not as a single propositional formula but as an inequality between the scores  $\mu$  (computed e.g., by **WMC**) of two independent propositional formulas  $\mu(P_w) > \mu(P_l)$ .

To bridge this gap, we define **preference structure**, a relational structure and semantic encoding, that allows us to capture the semantics of DPA losses in a modular fashion using a *single* propositional formula coupled with auxiliary constraints. This structure, based on a novel construction in propositional logic, makes it easy to cleanly characterize different DPA losses. We will use it to generalize the semantic loss and create a novel logic for DPA.

**Preference structure.** A preference structure is a tuple  $\bar{P} = (P, P_C, P_A)$  that, as will become clear shortly from Prop 2, captures the semantics of a winner and a loser. It consists of three propositional formulas: a **core semantic formula**  $P$  coupled with **conditioning constraints**  $P_C$  (as in Eq 3, which restrict the propositional models that can be counted), and **additive constraints**  $P_A$  that tell us which propositional models must always be counted. As we will show, all 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), differing only in their constraints ( $P_C$  and  $P_A$ ).

Each preference structure has a **formula form**  $\bar{P}_f$  and a **negated formula form**  $\neg\bar{P}_f$ , defined as follows:

$$\bar{P}_f := (P \vee P_A) \wedge P_C, \quad \neg\bar{P}_f := (\neg P \vee P_A) \wedge P_C. \quad (5)$$

Intuitively,  $\bar{P}_f$  and  $\neg\bar{P}_f$  correspond to the semantics of the winner ( $P_w$ ) and the loser ( $P_l$ ), respectively. Preference structures and their corresponding formula forms are designed to give us a modular way to express the original semantic loss, the conditional semantic loss, and arbitrary pairwise preferences. For example, removing  $P_A$  or making it  $\perp$  makes the semantic loss of  $\bar{P}_f$  equivalent to the conditional semantic loss from Eq 3. For convenience later, we note the two such equivalences formally below.

**Observation 2** (no conditioning or additive constraints). *When a preference structure  $\bar{P}$  has  $P_A$  and  $P_C$  set to  $\perp$  (false) and  $\top$  (true), respectively, the semantic loss of  $\bar{P}$  is equal to the standard semantic loss of  $\bar{P}_f$ , or  $\ell(P, \theta, D) = \ell(\bar{P}_f, \theta, D)$  (under Eq 4) given the logical equivalence of  $P$  and  $\bar{P}_f$ .*

**Observation 3** (no additive constraints). *When a preference structure  $\bar{P}$  has  $P_A$  set to  $\perp$ , the conditional semantic loss (Eq 3) of  $\bar{P}$  conditioned on  $P_C$ , or  $\ell(P \mid P_C, \theta, D)$ , is equal to  $\mathbb{E}_{d \sim D} \left[ -\log \sigma \left( \frac{\text{WMC}(\bar{P}_{f_d}, \theta)}{\text{WMC}(\neg\bar{P}_{f_d}, \theta)} \right) \right]$  given the equivalence with the conditional form in Eq. 4.*

With full preference structures containing  $P_A$ , any two propositional formulas (e.g., any  $P_w$  and  $P_l$ ) can be expressed as a preference structure based on a particular construction, called the **implication form**, which will play a central role when doing decompilation in Section 5.2.

**Proposition 2.** *Given any two propositional formulas  $P_w$  and  $P_l$ , there exists a preference structure  $\bar{P}$  such that  $P_w \equiv \bar{P}_f$  and  $P_l \equiv \neg\bar{P}_f$ .*

*Proof.* We provide a specific construction called the **implication form** of  $P_w$  and  $P_l$ , based on the following logical

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

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

equivalences, which can be checked manually:

$$P_w \equiv \left( \underbrace{(P_l \rightarrow P_w)}_P \vee \underbrace{(P_w \wedge P_l)}_{P_A} \right) \wedge \underbrace{(P_w \vee P_l)}_{P_C}$$

$$P_l \equiv \left( \underbrace{\neg(P_l \rightarrow P_w)}_{\neg P} \vee \underbrace{(P_w \wedge P_l)}_{P_A} \right) \wedge \underbrace{(P_w \vee P_l)}_{P_C}$$

As noted above, this construction corresponds exactly to the preference structure  $(P, P_C, P_A)$  with  $P := P_l \rightarrow P_w$ ,  $P_C := P_w \vee P_l$  and  $P_A := P_w \wedge P_l$ . and its two formula forms. (As a special case, whenever  $P_l \equiv \neg P_w$ , this simplifies to the structure  $\bar{P} = (P_w, \top, \perp)$ ; see again Obs 2.)  $\square$

As a corollary, this tell us that we can decompose any preference structure formed via the implication form to two formulas. Figure 3 shows a natural encoding of preference structures as Boolean truth tables (where a  $\checkmark$  denotes whether the corresponding model is counted in  $P_w$  and a  $\times$  denotes whether it is counted in  $P_l$ ), which we will later use when discussing and introducing new losses.

**Example 5** (preference structures and Boolean representations). *The truth table representations in Figure 3 have a natural mapping to preference structures. For example, the column for  $\ell_{\text{CPO}}$  can be expressed as two formulas:  $\mathbf{M}(\mathbf{x}, \mathbf{y}_w)$  (for  $\checkmark$ ) and  $\mathbf{M}(\mathbf{x}, \mathbf{y}_l)$  (for  $\times$ ), which can be compiled into a preference structure using the implication construction with  $P := \mathbf{M}(\mathbf{x}, \mathbf{y}_l) \rightarrow \mathbf{M}(\mathbf{x}, \mathbf{y}_w)$ ,  $P_C := \mathbf{M}(\mathbf{x}, \mathbf{y}_l) \vee \mathbf{M}(\mathbf{x}, \mathbf{y}_w)$  and  $P_A := \mathbf{M}(\mathbf{x}, \mathbf{y}_l) \wedge \mathbf{M}(\mathbf{x}, \mathbf{y}_w)$ . Visually,  $P_C$  corresponds to the union of rows containing a  $\checkmark$ s or  $\times$ s, and  $P_A$  to all rows with both  $\checkmark$ s and  $\times$ s. In relation to Obs 2- 3, we can say intuitively that columns where the original semantic loss is applicable are ones where all rows have a single mark, and columns where no double marks are included can be modeled using the conditional semantic loss (Eq 3).*

## 5.1. Semantic loss based on preference structures

In our generalization of the semantic loss, formulas  $P$  will be replaced with preference structures  $\bar{P}$ . For example, we can modify the logistic log form of SL in Eq 4 to be  $\ell(\bar{P}, \theta, D)$  and change the semantic loss ratio  $\rho_{\text{sem}}$  accordingly to operate over the formula forms of  $\bar{P}$  in Eq 5. 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:**  $\ell_{sl}(\bar{P}, \theta, D) := \mathbb{E}_{d \sim D}[f(\rho_{sem}(d), \beta)]$  where, like in DPA, different choices can be made about what  $f$  to apply over the semantic loss ratio  $\rho_{sem}$ , which gives rise to novel logics (we describe such variants in Table 1). To match the structure of DPA, we also add a weight parameter  $\beta$ .

Given Observations 2 and 3, we can see how the standard and conditional semantic loss end up being special cases of  $\ell_{sl-log}$  under specific preference structures and when  $\beta = 1$ .

**How many loss functions are there?** Under this formulation, we can view loss creation as a generative procedure: select an  $f$  then sample two formulas  $P_w$  and  $P_l$  (each denoting a unique Boolean function in  $n$  variables) to create a  $\bar{P}$  via Prop 2 (see also Figure 3). Absent any constraints, the total number of definable preference structures is doubly exponential in the number of probabilistic predictions  $n$ , specifically  $4^{2^n}$  (i.e., all unique pairs of Boolean functions). While not all such preference structures will lead to meaningful or unique losses, for DPO ( $n = 4$ ), this results in an upper bound of about 4.3 billion definable losses.

In particular, we will later refer to **non-trivial losses** as those where each  $P_w$  and  $P_l$  are not equal to one another or equal to  $\perp$  (in the truth table representations, these would be cases where the set of  $\checkmark$ s is identical to the set of  $\times$ s or there are no  $\checkmark$ s or  $\times$ s).

**How is the loss space structured?** While the space is large, one can structure this space using the semantics of the corresponding formulas. Below we define preference structure *entailment* and *equivalence*, and relate these semantic notions to the behavior of the compiled losses. These formal notions not only give us tools for structuring the DPA loss space but also inform the search for new loss functions.

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

**Proposition 3** (monotonicity). *If  $\bar{P}^{(1)} \subseteq \bar{P}^{(2)}$  then  $\ell_{sl}(\bar{P}^{(1)}, \theta, D) \geq \ell_{sl}(\bar{P}^{(2)}, \theta, D)$  for any  $\theta, D$ .*

We will later use entailment to characterize the relative strength of DPA losses and visualize their relations using a representation called a **loss lattice** (see Figure 5). We also extend entailment to **preference equivalence**  $\bar{P}^{(1)} \equiv \bar{P}^{(2)}$  in the natural way, namely when  $\bar{P}^{(1)} \subseteq \bar{P}^{(2)}$  and  $\bar{P}^{(2)} \subseteq \bar{P}^{(1)}$ . Equivalent preference structures have identical semantic losses (see Corollary 1 in Appendix D).

**Example 6** (loss entailment). *Entailments can be observed using the truth table encodings for preference structures  $\bar{P}_x$*

*as in Figure 3 and checking for subset relations between  $\checkmark$ s and  $\times$ s as in ordinary logic. For example, we can see that  $\ell_{unCPO}$  is (strictly) entailed by both  $\ell_{ORPO}$  and  $\ell_{CPO}$  by seeing that the  $\checkmark$ s of the latter are contained in the former and that the  $\times$ s of the former are contained in the latter. This will allow us to prove the following kinds of results about specific losses (where  $\bar{P}_x$  corresponds to the preference structure and semantic encoding for each loss  $x$ ):*

**Proposition 4.**  $\forall D, \theta. \ell_{sl}(\bar{P}_{CPO}, \theta, D) > \ell_{sl}(\bar{P}_{unCPO}, \theta, D)$  and  $\ell_{sl}(\bar{P}_{ORPO}, \theta, D) > \ell_{sl}(\bar{P}_{unCPO}, \theta, D)$ .

*and gives us a formal tool for thinking about the relative constrainedness of losses in a way that is grounded both in the semantics of those losses and their relative loss behavior. Through formalization, we can also reason about the relationship between new losses and standard losses such as cross-entropy  $\ell_{CE}$  (all these properties can be visualized in the kinds of lattices we show in Figure 5).*

## 5.2. Decompiling DPA losses into preference structures

The **decompilation** of a DPA loss  $\ell_{DPA_x}$  into a symbolic form can now be stated as finding a preference structure  $\bar{P}$  whose particular semantic loss  $\ell_{sl_x}$  is equal to  $\ell_{DPA_x}$ :

$$\underbrace{\ell_{DPA_x}(\theta, D) = \ell_{sl_x}(\bar{P}, \theta, D)}_{\text{decompilation of } \ell_{DPA_x} \text{ to } \bar{P}}, \quad \frac{\rho_{\theta}^t}{\rho_{\theta}^b} = \frac{\text{WMC}(\bar{P}_f; \theta)}{\text{WMC}(\neg \bar{P}_f; \theta)} \quad (6)$$

We say that a preference structure  $\bar{P}$  **correctly characterizes** a loss  $\ell_x$  under some  $\ell_{sl_x}$  whenever this condition holds. Given the structure of the DPA loss (Eq 1) and the generalized semantic loss, whenever  $f$  is fixed this can be reduced to finding a  $\bar{P}$  whose semantic loss ratio  $\rho_{sem}$  is equal to  $\ell_x$ 's core loss equation  $\rho_{\theta}$  as shown on the right of Eq 6 (with the log removed).

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

**Characterizing the DPA equation class.** By construction, we will assume that all the core equations for DPA losses  $\rho_{\theta}^t$  and  $\rho_{\theta}^b$  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  $J_i \subseteq \{1, \dots, n\}$  such that  $e_i = \prod_{j \in J_i} \ell_{ij}$  where  $\ell_{ij}$  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 multilinear



**Algorithm 1** Translation of loss to logic (decompilation)

**input** Disjoint polynomial  $\rho_\theta = \log \frac{\rho_\theta^t}{\rho_\theta^b}$  **output**  $\bar{P}$   
 $P_t \leftarrow \text{SEM}(\rho_\theta^t)$  {Translation to logic, Table 7}  
 $P_b \leftarrow \text{SEM}(\rho_\theta^b)$   
 $P \leftarrow \text{SIMPLIFY}(\text{Implies}(P_b, P_t))$  {Implication form}  
 $P_C \leftarrow \text{SIMPLIFY}(\text{Or}(P_t, P_b))$  {via Proposition 2}  
 $P_A \leftarrow \text{SIMPLIFY}(\text{And}(P_t, P_b))$   
**return**  $\bar{P} := (P, P_C, P_A)$  { $\rho_\theta = \log \frac{\text{WMC}(\bar{P}_f; \theta)}{\text{WMC}(\bar{P}_f; \theta)}$ , Lem. 1}

form, including the original form of DPOP (Pal et al., 2024) which we discuss in Appendix G and fix through **variable copying** as shown in Table 2.

**Translation algorithm.** Our translation process is shown in Algorithm 1. Given  $\rho_\theta$ , both  $\rho_\theta^t$  and  $\rho_\theta^b$  are independently translated into logic via a compositional translation function SEM. The translation is standard, based on the rules in Table 7: first each model prediction  $P_{\mathbf{M}}(\cdot)$  is mapped to a probabilistic prediction  $\mathbf{M}(\cdot)$ ; then  $1 - P$  is mapped to negation,  $P_1 \cdot P_2$  to conjunction, and  $P_1 + P_2$  to disjunction; these rules are applied repeatedly until the full expression is translated. By induction on the rules, one can establish the correctness of the translation function SEM, i.e., that for any disjoint multilinear polynomial  $\rho_\theta^z$ , it holds that  $\rho_\theta^z = \text{WMC}(\text{SEM}(\rho_\theta^z); \theta)$ . Finally, the implication construction from Prop 2 is applied to create a preference structure  $\bar{P}$ , where formulas are (optionally) minimized via SIMPLIFY.

The following follows from the correctness of our translation rules and the implication construction (Prop 2):

**Lemma 1** (correctness of translation). *Given a loss equation  $\rho_\theta = \log \rho_\theta^t / \rho_\theta^b$  with disjoint multilinear polynomials  $\rho_\theta^t$ , and  $\rho_\theta^b$ , Algorithm 1 returns a preference structure  $\bar{P}$  whose semantic loss ratio  $\rho_{sem}$  equals  $\rho_\theta$ , satisfying Eq 6 (right).*

This establishes the correctness of our decompilation algorithm, showing specifically that Algorithm 1 yields preference structures that satisfy the right equality in Eq 6.

**Example 7** (loss derivation). *Figure 4 shows an example derivation of the original  $\ell_{\text{ORPO}}$  (Hong et al., 2024) with  $\text{ODDS}_\theta(y | x) := \frac{P_\theta(y|x)}{1-P_\theta(y|x)}$  into a preference structure. First, the loss is reduced to its core loss equation  $\rho_\theta$  as in Table 2 (with the log removed). Parts of this equation are then compositionally translated to logic via Algorithm 1, with  $\rho_\theta^t$  corresponding to  $\mathbf{M}(x, y_w) \wedge \neg \mathbf{M}(x, y_l)$  and  $\rho_\theta^b$  to  $\mathbf{M}(x, y_l) \wedge \neg \mathbf{M}(x, y_w)$ . A preference structure is then constructed by making  $P := (\mathbf{M}(x, y_l) \wedge \neg \mathbf{M}(x, y_w)) \rightarrow (\mathbf{M}(x, y_w) \wedge \neg \mathbf{M}(x, y_l))$  (which is logically equivalent to  $\mathbf{M}(x, y_l) \rightarrow \mathbf{M}(x, y_w)$ ),  $P_A := \perp$  (after simplification) and  $P_C := (\mathbf{M}(x, y_w) \wedge \neg \mathbf{M}(x, y_l)) \vee (\neg \mathbf{M}(x, y_w) \wedge \mathbf{M}(x, y_l))$  (see Figure 9 for an example of how to compute this using symbolic computation tools).*

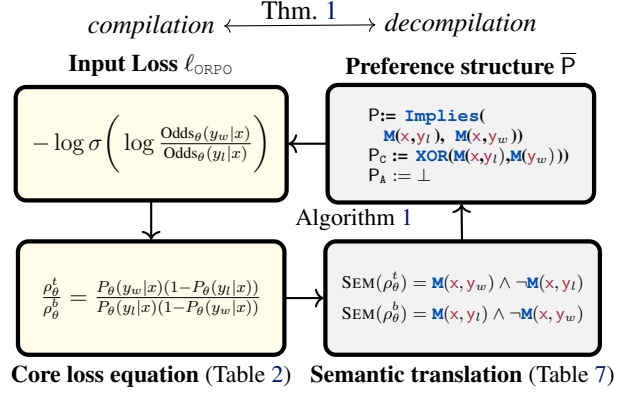


Figure 4. How do we decompile losses? A visualization of our compositional decompilation procedure and main results using the example loss  $\ell_{\text{ORPO}}$ . First the original **input loss** (upper left) is stripped down to its **core loss equation** (lower left, log removed), which is then **semantically translated** (lower right) and mapped into a **preference structure** (upper right) that can be **compiled** back into the original loss (Thm 1).

## 6. Results and Discussion

Table 4 shows the preference structures obtained from Algorithm 1 for the DPA losses in Table 2. The following result establishes their correctness:

**Theorem 1** (Correctness of formalization). *The preference structures in Table 4 correctly characterize the losses in Table 2 and satisfy Eq 6 under semantic loss  $\ell_{\text{sl-log}}$  (Table 3).*

*Proof.* Since the original losses were all formulated using the logistic log form of DPA, the correctness of Algorithm 1 (which follows from Lemma 1) implies that compiling the representations in Table 4 (which, as noted above, were obtained by running Algorithm 1 on the losses in Table 2) under  $\ell_{\text{sl-log}}$  will yield precisely the original losses, and hence satisfies Eq 6.  $\square$

By changing the version of semantic loss, we can extend our analysis to other variants of DPO, showing the generality of our semantic analysis and its invariance to the choice of  $f$ . For example, by changing  $\ell_{\text{sl-log}}$  to  $\ell_{\text{sl-squared}}$  or  $\ell_{\text{sl-margin}}$ , we immediately obtain the following:

**Theorem 2** (Extension to other DPOs). *The DPO and CPO preference structures in Table 4 correctly characterize the IPO and SLIC losses (Table 1) and satisfy Eq 6 under the  $\ell_{\text{sl-squared}}$  and  $\ell_{\text{sl-margin}}$  semantic losses, respectively.*

Interestingly, we show in Appendix H how perceptron-style losses such as RRHF in Table 1 can be derived from our representations by changing our underlying logic to fuzzy logic, which is a popular alternative to our probabilistic logic

Loss	Representation $\bar{P}$
CE	$P := \mathbf{M}(x, y_w), P_C := \perp$
CEUnl	$P := \text{And}(\mathbf{M}(x, y_w), \text{Not}(\mathbf{M}(x, y_l)))$ $P_C := \perp$
CPO	$;;$ core semantic formula $P := \text{Implies}(\mathbf{M}(x, y_l), \mathbf{M}(x, y_w))$ $;;$ one-true constraint $P_C := \text{Or}(\mathbf{M}(x, y_l), \mathbf{M}(x, y_w))$
ORPO	$P := \text{Implies}(\mathbf{M}(x, y_l), \mathbf{M}(x, y_w))$ $;;$ one-hot constraint $P_C := \text{XOR}(\mathbf{M}(x, y_l), \mathbf{M}(x, y_w))$
DPO	$;;$ reference form of CPO $P := \text{Implies}(\text{And}(\text{Ref}(x, y_w), \mathbf{M}(x, y_l)), \text{And}(\text{Ref}(x, y_l), \mathbf{M}(x, y_w)))$ $P_C := \text{Or}(\text{And}(\text{Ref}(x, y_w), \mathbf{M}(x, y_l)), \text{And}(\text{Ref}(x, y_l), \mathbf{M}(x, y_w)))$
SimPO	$;;$ DPO with manual reference policy $P := \text{Implies}(\text{And}(\text{Mref}(x, y_w), \mathbf{M}(x, y_l)), \text{And}(\text{Mref}(x, y_l), \mathbf{M}(x, y_w)))$ $P_C := \text{Or}(\text{And}(\text{Mref}(x, y_w), \mathbf{M}(x, y_l)), \text{And}(\text{Mref}(x, y_l), \mathbf{M}(x, y_w)))$

Table 4. What do formalized versions of standard losses look like? Formalizations of some of the losses from Table 2 shown in terms of  $P$  and  $P_C$  (for succinctness, we exclude  $P_A$  which can be inferred from each  $P_C$  via Algorithm 1).

approach. Given the ubiquity of DPO-style updates in other online variants of DPA (Qi et al., 2024; Zhang et al., 2024; Chen et al., 2024b; Guo et al., 2024), we also believe that our semantic analysis might also be useful for semantically characterizing online learning approaches, which we see as a promising future direction of research.

### 6.1. What do we learn about known losses?

**Single model approaches have an intuitive semantics and are 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  $P_C$  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 Figures 5- 6, we see that both are entailed by the CEUnl baseline, yet have a non-entailing relation to one another and the cross-entropy loss.

In general, we see that preference losses are highly constrained. This is in contrast to the losses typically used with the semantic loss, suggesting that there is much to

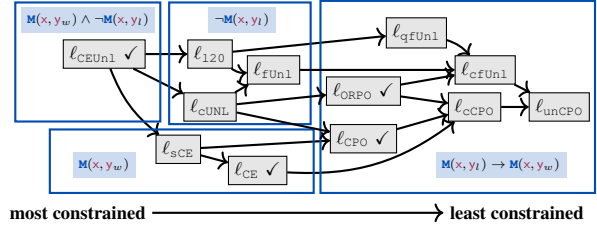


Figure 5. What other losses are there? Here we show the loss landscape for single model preference approaches using a **loss lattice** showing losses (nodes) structured according to strict entailment ( $\square$ ) and their core formulas  $\bar{P}$  (boxes) with  $\checkmark$  being the known losses. See Appendix E for details of the individual losses and a more exhaustive lattice with DPO variants in Figure 6.

learn by working backward from empirically successful loss functions to their semantic properties.

**Example 8 (unconstrained loss).** The loss  $\ell_{\text{unCPO}}$  is an unconstrained version of CPO, with  $P := \mathbf{M}(x, y_l) \rightarrow \mathbf{M}(x, y_w)$  and  $P_C := \top, P_A := \perp$  (see again semantics in Figure 3), and is typical of the kinds of losses produced by the standard semantic loss. When compiled into a loss, this leads to the rather cumbersome core loss equation:  $\log \frac{P_\theta(y_l|x)P_\theta(y_w|x) + (1 - P_\theta(y_l|x))}{P_\theta(y_l|x)(1 - P_\theta(y_w|x))}$ , which is a loss that would be very hard to derive working from DPO. While this loss has issues that we discuss Appendix F.1, it can be completely justified semantically, and we believe that this shows the value of having a reliable semantic framework for deriving new losses that would be otherwise difficult to derive from existing DPA approaches.

**There are many losses still to explore and we can exhaustively enumerate them.** We created new losses by modifying the conditioning constraints of existing losses. Figure 5 shows a (non-exhaustive) lattice representation of the loss landscape for single model preference approaches created by mechanically deriving new losses from the  $\ell_{\text{CEUnl}}$  baseline (the most constrained) and ordering them by strict entailment (terminating in  $\ell_{\text{unCPO}}$ , our running example). We see different **semantic regions** emerge characterized by different formulas  $P$ , notably an unexplored region of unlikelihood losses ( $\ell_{120}, \ell_{\text{cUNL}}, \ell_{\text{fUNL}}$ ) that optimize for the negation of the loser  $\neg \mathbf{M}(x, y_l)$ . Through compilation, any of these losses are now subject to experimentation.

In Figure 6, we show an extended version of Figure 5 with the reference forms (gray boxes) of all losses (discussed below). Keeping the variables constant, this version exhaustively captures all definable non-trivial single model losses and preference structures  $\bar{P}$  (nodes in this graph) that lie semantically in-between the baseline  $\ell_{\text{CEUnl}}$  and  $\ell_{\text{unCPO}}$ , or formally all  $\bar{P}$  s.t.  $\bar{P}_{\text{CEUnl}} \subseteq \bar{P} \subseteq \bar{P}_{\text{unCPO}}$ . We conjecture that this class of losses captures the most promising single model alternatives to the known losses  $\ell_{\text{CPO}}$  and  $\ell_{\text{ORPO}}$  and

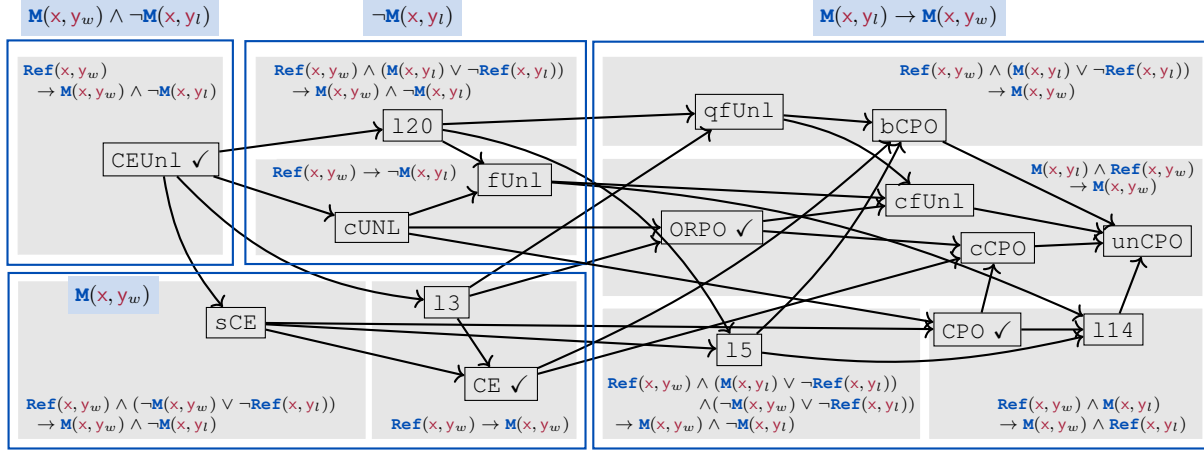


Figure 6. What are interesting DPO variants to explore? Extending the loss lattice in Figure 5 to a version of the single model losses with reference models (i.e., their **reference forms**), showing different (largely unexplored) variants of DPO and the different semantics regions (gray boxes, corresponding to the core semantic formula for P each set of losses). See Appendix E for details.

empirically investigate some of these losses below.

**Adding a reference model has a clear, though sometimes peculiar, semantics.** *What happens semantically when we add a reference model or term to our loss?* As discussed in Section 3 and Obs. 1, specific losses can be made into a DPO-style losses with reference information (i.e., the reference form of that loss) by subtracting the log ratio  $s_{\text{ref}}(y_w, y_l)$  from that loss’s core loss equation. The following proposition shows the semantic result of adding reference information and follows directly from Obs. 1 and the application of Algorithm 1.

**Proposition 5** (semantics of reference forms). *Given a loss characterized by the core loss equation  $\rho_\theta$  equal to  $\log \rho_\theta^t / \rho_\theta^b$ , the core semantic formula P for that loss’s reference form is logically equivalent to the formula  $(\text{SEM}(\rho_\theta^b) \wedge \text{Ref}(x, y_w)) \rightarrow (\text{SEM}(\rho_\theta^t) \wedge \text{Ref}(x, y_l))$ .*

The semantics of DPO, which is the reference form of CPO, is shown in Table 4 and is logically equivalent to a conjunction of two implications:  $\text{Ref}(x, y_w) \wedge M(x, y_l) \rightarrow M(x, y_w)$  and  $\text{Ref}(x, y_w) \wedge \neg \text{Ref}(x, y_l) \rightarrow \neg M(x, y_l)$ . 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.

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 (see Appendix G). 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. Several such variants of DPO and SimPO are shown in Figure 6 (i.e., the gray regions).

**Example 9** (Novel variant of cross-entropy). *In addition to finding novel variants of DPO, the reference forms can also reveal intriguing variants of standard losses like cross-entropy  $\ell_{\text{CE}}$ . Semantically,  $\ell_{\text{CE}}$  can be expressed in an implication form as  $\neg M(x, y_w) \rightarrow M(x, y_w)$ . By adding a reference model according to Prop. 5, this results in the formula  $(\neg M(x, y_w) \wedge \text{Ref}(x, y_w)) \rightarrow M(x, y_w) \wedge \text{Ref}(x, y_l)$ , which simplifies to the logically equivalent formula  $\text{Ref}(x, y_w) \rightarrow M(x, y_w)$ . Semantically, this leads to a variant of cross-entropy where updates are made based on signal from the reference model, which seems like a natural variation. The full loss equation results in  $\rho_\theta = \log \frac{P_\theta(y_w|x)P_{\text{ref}}(y_l|x)}{(1-P_\theta(y_w|x))P_{\text{ref}}(y_w|x)}$ ; by modifying the conditioning constraints one can arrive at different variants of this loss and directly implement each variant for experimentation.*

**Can we find empirically improved losses using our method?** The ultimate goal of our analysis is to facilitate the discovery of empirically improved DPA losses. As a case study, we implemented single model losses around the known  $\ell_{\text{CPO}}$  in Figure 5, treating it as a baseline to improve upon. Using a model-as-judge style evaluation from Hong et al. (2024) and a Qwen-0.5B LLM (details in Appendix F), we found one particular loss,  $\ell_{\text{CCPO}}$  to be competitive with  $\ell_{\text{CPO}}$ , achieving a win-rate of 52.0 as shown in Table 5. We also observe that different losses have markedly

different performance across different datasets, suggesting that a one-size-fits-all approach isn't ideal—semantically different tasks are best learned using different losses.

loss	WR% ( $\ell_{\text{CPO}}$ )	evol	false-qa	flan	sharegpt	ultrachat
$\ell_{\text{CEUNL}}$	46.1 ( $\pm 0.4$ )	46.1 ( $\pm 2.2$ )	51.6 ( $\pm 2.9$ )	46.4 ( $\pm 1.7$ )	46.2 ( $\pm 1.2$ )	44.1 ( $\pm 1.0$ )
$\ell_{\text{GEUNL}}$	48.9 ( $\pm 0.8$ )	45.3 ( $\pm 1.9$ )	34.7 ( $\pm 6.3$ )	57.9 ( $\pm 1.2$ )	46.8 ( $\pm 2.4$ )	41.3 ( $\pm 1.4$ )
$\ell_{\text{ECPO}}$	52.0 ( $\pm 0.6$ )	50.7 ( $\pm 0.5$ )	50.2 ( $\pm 0.7$ )	57.2 ( $\pm 1.1$ )	47.2 ( $\pm 1.8$ )	53.1 ( $\pm 1.9$ )
$\ell_{\text{UNCPO}}$	46.0 ( $\pm 0.2$ )	45.8 ( $\pm 0.3$ )	52.1 ( $\pm 3.0$ )	45.7 ( $\pm 0.6$ )	46.2 ( $\pm 2.1$ )	44.8 ( $\pm 2.1$ )

Table 5. Results of a feasibility study involving Qwen-0.5B tuned on the new losses (rows) compared against the known loss  $\ell_{\text{CPO}}$  (second column) on *ultrafeedback* test in aggregate (2nd column) and on subsets (right columns). See details in Section F.

While small scale, this study demonstrates the feasibility of using our framework to derive empirically successful losses. Appendix F reports additional experiments and findings.

## 7. Conclusion

Despite the routine use of a variety of DPA algorithms to align LLMs with human preferences, knowing what exactly 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 a *decompilation* procedure, allowing one to compositionally 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.

## Acknowledgements

Special thanks to the following people for their feedback at various stages of the work (in alphabetical order): Kareem Ahmed, Gregor Betz, Junyan Cheng, Hamish Ivison, Maryna Kavalenka, Emile van Krieken, Nathan Lambert, Robin Manhaeve, Valentina Pyatkin, Antonio Vergari and Gijs Wijnholds, as well as the four anonymous reviewers who read an earlier draft (in particular, we thank one reviewer who provided useful details about standard WMC encodings and feedback related to Assumption 1). Thanks also to the audiences at Ai2, the University of Utah and the University of Stuttgart for listening to a talk version of this paper and for providing helpful feedback and discussion. As usual, any remaining mistakes remain our own.

## References

Ahmed, K., Teso, S., Chang, K.-W., Van den Broeck, G., and Vergari, A. Semantic probabilistic layers for neuro-

symbolic learning. *Advances in Neural Information Processing Systems*, 35:29944–29959, 2022.

Ahmed, K., Chang, K.-W., and Van den Broeck, G. A pseudo-semantic loss for deep generative models with logical constraints. *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023a.

Ahmed, K., Teso, S., Moretti, P., Di Liello, L., Ardino, P., Gobbi, J., Liang, Y., Wang, E., Chang, K.-W., Passerini, A., et al. Semantic loss functions for neuro-symbolic structured prediction. In *Compendium of Neurosymbolic Artificial Intelligence*, pp. 485–505. IOS Press, 2023b.

Amini, A., Vieira, T., and Cotterell, R. Direct preference optimization with an offset. *In Findings of ACL*, 2024.

Asai, A. and Hajishirzi, H. Logic-guided data augmentation and regularization for consistent question answering. *Proceedings of ACL*, 2020.

Azar, M. G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*, 2023.

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

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

Cai, Z., Cao, M., Chen, H., Chen, K., Chen, K., Chen, X., Chen, X., Chen, Z., Chen, Z., Chu, P., Dong, X., Duan, H., Fan, Q., Fei, Z., Gao, Y., Ge, J., Gu, C., Gu, Y., Gui, T., Guo, A., Guo, Q., He, C., Hu, Y., Huang, T., Jiang, T., Jiao, P., Jin, Z., Lei, Z., Li, J., Li, J., Li, L., Li, S., Li, W., Li, Y., Liu, H., Liu, J., Hong, J., Liu, K., Liu, K., Liu, X., Lv, C., Lv, H., Lv, K., Ma, L., Ma, R., Ma, Z., Ning, W., Ouyang, L., Qiu, J., Qu, Y., Shang, F., Shao, Y., Song, D., Song, Z., Sui, Z., Sun, P., Sun, Y., Tang, H., Wang, B., Wang, G., Wang, J., Wang, J., Wang, R., Wang, Y., Wang, Z., Wei, X., Weng, Q., Wu, F., Xiong, Y., Xu, C., Xu, R., Yan, H., Yan, Y., Yang, X., Ye, H., Ying, H., Yu, J., Yu, J., Zang, Y., Zhang, C., Zhang, L., Zhang, P., Zhang, P., Zhang, R., Zhang, S., Zhang, S., Zhang, W., Zhang, W., Zhang, X., Zhang, X., Zhao, H., Zhao, Q., Zhao, X., Zhou, F., Zhou, Z., Zhuo, J., Zou, Y., Qiu, X., Qiao, Y., and Lin, D. Internlm2 technical report, 2024.

Calanzone, D., Teso, S., and Vergari, A. Logically consistent language models via neuro-symbolic integration. *arXiv preprint arXiv:2409.13724*, 2024.



- Chavira, M. and Darwiche, A. On probabilistic inference by weighted model counting. *Artificial Intelligence*, 172 (6-7):772–799, 2008.
- Chen, A., Malladi, S., Zhang, L. H., Chen, X., Zhang, Q., Ranganath, R., and Cho, K. Preference learning algorithms do not learn preference rankings. *Proceedings of NeurIPS*, 2024a.
- Chen, Z., Deng, Y., Yuan, H., Ji, K., and Gu, Q. Self-play fine-tuning converts weak language models to strong language models. *Proceedings of ICML*, 2024b.
- Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
- Cui, G., Yuan, L., Ding, N., Yao, G., Zhu, W., Ni, Y., Xie, G., Liu, Z., and Sun, M. Ultrafeedback: Boosting language models with high-quality feedback, 2023.
- Dai, J., Pan, X., Sun, R., Ji, J., Xu, X., Liu, M., Wang, Y., and Yang, Y. Safe rlhf: Safe reinforcement learning from human feedback. In *Proceedings of ICLR*, 2024.
- De Raedt, L. and Kimmig, A. Probabilistic (logic) programming concepts. *Machine Learning*, 100:5–47, 2015.
- De Raedt, L., Kimmig, A., and Toivonen, H. Problog: A probabilistic prolog and its application in link discovery. In *Proceedings of IJCAI*, pp. 2462–2467, 2007.
- Dohan, D., Xu, W., Lewkowycz, A., Austin, J., Bieber, D., Lopes, R. G., Wu, Y., Michalewski, H., Sauros, R. A., Sohl-Dickstein, J., et al. Language model cascades. *arXiv preprint arXiv:2207.10342*, 2022.
- Donadello, I., Serafini, L., and Garcez, A. D. Logic tensor networks for semantic image interpretation. *arXiv preprint arXiv:1705.08968*, 2017.
- Eisner, J., Goldlust, E., and Smith, N. A. Dyna: A declarative language for implementing dynamic programs. In *Proc. of ACL*, 2004.
- Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. Kto: Model alignment as prospect theoretic optimization. *Proceedings of ICML*, 2024.
- Fierens, D., Van den Broeck, G., Renkens, J., Shterionov, D., Gutmann, B., Thon, I., Janssens, G., and De Raedt, L. Inference and learning in probabilistic logic programs using weighted boolean formulas. *Theory and Practice of Logic Programming*, 15(3):358–401, 2015.
- Fischer, M., Balunovic, M., Drachler-Cohen, D., Gehr, T., Zhang, C., and Vechev, M. DI2: training and querying neural networks with logic. In *Proceedings of ICML*, 2019.
- Friedman, D., Wettig, A., and Chen, D. Learning transformer programs. *Advances in Neural Information Processing Systems*, 36, 2024.
- Giannini, F., Marra, G., Diligenti, M., Maggini, M., and Gori, M. On the relation between loss functions and t-norms. In *Inductive Logic Programming: 29th International Conference*, 2020.
- Giannini, F., Diligenti, M., Maggini, M., Gori, M., and Marra, G. T-norms driven loss functions for machine learning. *Applied Intelligence*, 53(15):18775–18789, 2023.
- Grespan, M. M., Gupta, A., and Srikumar, V. Evaluating relaxations of logic for neural networks: A comprehensive study. *Proceedings of IJCAI*, 2021.
- Guo, S., Zhang, B., Liu, T., Liu, T., Khalman, M., Llinares, F., Rame, A., Mesnard, T., Zhao, Y., Piot, B., et al. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*, 2024.
- Halpern, J. Y., Harper, R., Immerman, N., Kolaitis, P. G., Vardi, M. Y., and Vianu, V. On the unusual effectiveness of logic in computer science. *Bulletin of Symbolic Logic*, 7(2):213–236, 2001.
- Hewitt, J., Hahn, M., Ganguli, S., Liang, P., and Manning, C. D. RNNs can generate bounded hierarchical languages with optimal memory. In *Proceedings of EMNLP*, 2020.
- Hinnerichs, T., Manhaeve, R., Marra, G., and Dumancic, S. Declarative design of neural predicates in neuro-symbolic systems. *arXiv preprint arXiv:2405.09521*, 2024.
- Hong, J., Lee, N., and Thorne, J. Reference-free monolithic preference optimization with odds ratio. *Proceedings of EMNLP*, 2024.
- Hu, X., He, T., and Wipf, D. New desiderata for direct preference optimization. *arXiv preprint arXiv:2407.09072*, 2024.
- Iverson, H., Wang, Y., Pyatkin, V., Lambert, N., Peters, M., Dasigi, P., Jang, J., Wadden, D., Smith, N. A., Beltagy, I., et al. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*, 2023.
- Jeffrey, R. C. *The logic of decision*. University of Chicago press, 1965.
- Ji, J., Liu, M., Dai, J., Pan, X., Zhang, C., Bian, C., Chen, B., Sun, R., Wang, Y., and Yang, Y. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36, 2024.

- Khattab, O., Singhvi, A., Maheshwari, P., Zhang, Z., Santhanam, K., Vardhamanan, S., Haq, S., Sharma, A., Joshi, T. T., Moazam, H., et al. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.
- Klement, E. P., Mesiar, R., and Pap, E. *Triangular norms*, volume 8. Springer Science & Business Media, 2013.
- Li, T., Gupta, V., Mehta, M., and Srikumar, V. A Logic-Driven Framework for Consistency of Neural Models. In *Proceedings of EMNLP*, 2019.
- Li, Z., Huang, J., and Naik, M. Scallop: A language for neurosymbolic programming. *Proceedings of the ACM on Programming Languages*, 7(PLDI):1463–1487, 2023.
- Liu, W., Zeng, W., He, K., Jiang, Y., and He, J. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning. *arXiv preprint arXiv:2312.15685*, 2023.
- Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., and De Raedt, L. Deepprolog: Neural probabilistic logic programming. *Advances in neural information processing systems*, 31, 2018.
- Marconato, E., Teso, S., Vergari, A., and Passerini, A. Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts. *Advances in Neural Information Processing Systems*, 36, 2024.
- Marra, G., Giannini, F., Diligenti, M., and Gori, M. 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.
- Marra, G., Dumančić, S., Manhaeve, R., and De Raedt, L. From statistical relational to neurosymbolic artificial intelligence: A survey. *Artificial Intelligence*, pp. 104062, 2024.
- McCarthy, J. et al. *Programs with common sense*. RLE and MIT computation center Cambridge, MA, USA, 1960.
- McCulloch, W. S. and Pitts, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, 1943.
- Meng, Y., Xia, M., and Chen, D. Simpo: Simple preference optimization with a reference-free reward. *Proceedings of Neurips*, 2024.
- Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J. K., Singh, S., et al. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, 2017.
- Minervini, P. and Riedel, S. Adversarially regularising neural nli models to integrate logical background knowledge. *arXiv preprint arXiv:1808.08609*, 2018.
- Miranda, L. J. V., Wang, Y., Elazar, Y., Kumar, S., Pyatkin, V., Brahman, F., Smith, N. A., Hajishirzi, H., and Dasigi, P. Hybrid preferences: Learning to route instances for human vs. ai feedback. *arXiv preprint arXiv:2410.19133*, 2024.
- Nilsson, N. J. Logic and artificial intelligence. *Artificial intelligence*, 47(1-3):31–56, 1991.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- Pal, A., Karkhanis, D., Dooley, S., Roberts, M., Naidu, S., and White, C. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.
- Park, R., Rafailov, R., Ermon, S., and Finn, C. Disentangling length from quality in direct preference optimization. *Proceedings of ACL*, 2024.
- Qi, B., Li, P., Li, F., Gao, J., Zhang, K., and Zhou, B. Online dpo: Online direct preference optimization with fast-slow chasing. *arXiv preprint arXiv:2406.05534*, 2024.
- Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Proceedings of NeurIPS*, 2023.
- Razin, N., Malladi, S., Bhaskar, A., Chen, D., Arora, S., and Hanin, B. Unintentional unalignment: Likelihood displacement in direct preference optimization. *arXiv preprint arXiv:2410.08847*, 2024.
- Rescher, N. Semantic foundations for the logic of preference. *The logic of decision and action*, pp. 37–62, 1967.
- Richardson, K. and Wijnholds, G. Lectures on language model programming, August 2024. URL [https://github.com/yakazimir/esslli\\_2024\\_llm\\_programming](https://github.com/yakazimir/esslli_2024_llm_programming).
- Rocktäschel, T., Singh, S., and Riedel, S. Injecting logical background knowledge into embeddings for relation extraction. In *Proceedings of NAACL*, 2015.
- Saeidi, A., Verma, S., and Baral, C. Insights into alignment: Evaluating dpo and its variants across multiple tasks. *arXiv preprint arXiv:2404.14723*, 2024.

- Ślusarz, N., Komendantskaya, E., Daggett, M. L., Stewart, R., and Stark, K. Logic of differentiable logics: Towards a uniform semantics of dl. *arXiv preprint arXiv:2303.10650*, 2023.
- Stoy, J. E. *Denotational semantics: the Scott-Strachey approach to programming language theory*. MIT press, 1977.
- Tang, Y., Guo, Z. D., Zheng, Z., Calandriello, D., Munos, R., Rowland, M., Richemond, P. H., Valko, M., Pires, B. Á., and Piot, B. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*, 2024.
- van Krieken, E., Acar, E., and van Harmelen, F. Analyzing differentiable fuzzy logic operators. *Artificial Intelligence*, 302:103602, 2022.
- van Krieken, E., Badreddine, S., Manhaeve, R., and Giunchiglia, E. Uller: A unified language for learning and reasoning. In *International Conference on Neural-Symbolic Learning and Reasoning*, pp. 219–239. Springer, 2024a.
- van Krieken, E., Minervini, P., Ponti, E. M., and Vergari, A. On the independence assumption in neurosymbolic learning. *arXiv preprint arXiv:2404.08458*, 2024b.
- Vieira, T., Francis-Landau, M., Filardo, N. W., Khorasani, F., and Eisner, J. 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.
- von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., Huang, S., Rasul, K., and Gallowédec, Q. Trl: Transformer reinforcement learning. <https://github.com/huggingface/trl>, 2020.
- Wang, Y., Zhong, W., Li, L., Mi, F., Zeng, X., Huang, W., Shang, L., Jiang, X., and Liu, Q. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*, 2023.
- Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*, 2019.
- Winata, G. I., Zhao, H., Das, A., Tang, W., Yao, D. D., Zhang, S.-X., and Sahu, S. Preference tuning with human feedback on language, speech, and vision tasks: A survey. *arXiv preprint arXiv:2409.11564*, 2024.
- Xu, H., Sharaf, A., Chen, Y., Tan, W., Shen, L., Van Durme, B., Murray, K., and Kim, Y. J. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*, 2024.
- Xu, J., Zhang, Z., Friedman, T., Liang, Y., and Broeck, G. A Semantic Loss Function for Deep Learning with Symbolic Knowledge. In *Proceedings of ICML*, pp. 5498–5507, 2018.
- Yixing, L., Yuxian, G., Dong, L., Wang, D., Cheng, Y., and Wei, F. Direct preference knowledge distillation for large language models. In *arXiv preprint arXiv:2406.19774*, 2024.
- Yuan, Z., Yuan, H., Tan, C., Wang, W., Huang, S., and Huang, F. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023.
- Zadeh, L. A. Fuzzy logic and approximate reasoning. *Synthese*, 30(3):407–428, 1975.
- Zhang, S., Yu, D., Sharma, H., Zhong, H., Liu, Z., Yang, Z., Wang, S., Hassan, H., and Wang, Z. Self-exploring language models: Active preference elicitation for online alignment. *arXiv preprint arXiv:2405.19332*, 2024.
- Zhao, H., Winata, G. I., Das, A., Zhang, S.-X., Yao, D. D., Tang, W., and Sahu, S. Rainbowpo: A unified framework for combining improvements in preference optimization. *Proceedings of ICLR*, 2025.
- Zhao, Y., Khalman, M., Joshi, R., Narayan, S., Saleh, M., and Liu, P. J. Calibrating sequence likelihood improves conditional language generation. In *Proceedings of ICLR*, 2022.
- Zhao, Y., Joshi, R., Liu, T., Khalman, M., Saleh, M., and Liu, P. J. SLiC-HF: Sequence Likelihood Calibration with Human Feedback. *arXiv preprint arXiv:2305.10425*, 2023.
- Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

## A. Original losses

Further details of the original losses in Table 2, along with other variants such as R-DPO (Park et al., 2024), ODPO (Amini et al., 2024) and DPKD (Yixing et al., 2024), are shown in Table 6. While our formalization abstracts over certain details such as length normalization and additional regularization terms, we include such details from the original studies. In the case of regularization terms, as noted

## Understanding the Logic of Direct Preference Alignment through Logic

Loss name	core loss equation $\rho_\theta$	CE term	length norm.	Extra details and terms
<b>common baseline losses</b>				
$\ell_{\text{CE}}$	$\log \frac{P_\theta(y_w x)}{1-P_\theta(y_w x)}$	—	—	
$\ell_{\text{CEUnl}}$ (Rafailov et al., 2023)	$\log \frac{P_\theta(y_w x)(1-P_\theta(y_l x))}{1-(P_\theta(y_w x)(1-P_\theta(y_l x)))}$	—	—	Unlikelihood term weighted by $\alpha$
<b>reference approaches</b>				
$\ell_{\text{DPO}}$ (Rafailov et al., 2023)	$\log \frac{P_\theta(y_w x)P_{\text{ref}}(y_l x)}{P_{\text{ref}}(y_w x)P_\theta(y_l x)}$	×	×	
$\ell_{\text{ODPO}}$ (Amini et al., 2024)	$\log \frac{P_\theta(y_w x)P_{\text{ref}}(y_l x)}{P_{\text{ref}}(y_w x)P_\theta(y_l x)} - \gamma_{\text{offset}}$	×	×	Added offset term $\gamma_{\text{offset}}$
$\ell_{\text{DPOP}}$ (Pal et al., 2024)	$\log \frac{P_\theta(y_w x)P_{\text{ref}^2}(y_w x)P_{\text{ref}}(y_l x)}{P_{\text{ref}}(y_w x)P_{\text{ref}^2}(y_w x)P_\theta(y_l x)}$	×	×	See Appendix G
$\ell_{\text{R-DPO}}$ (Park et al., 2024)	$\log \frac{P_\theta(y_w x)P_{\text{ref}}(y_l x)}{P_{\text{ref}}(y_w x)P_\theta(y_l x)} + \gamma_{\text{len}}$	×	×	Added length bias term $\gamma_{\text{len}}$
$\ell_{\text{DPKD}}$ (Yixing et al., 2024)	$\log \frac{P_{\text{student}}(y_w x)P_{\text{teacher}}(y_l x)}{P_{\text{teacher}}(y_w x)P_{\text{student}}(y_l x)}$	✓	✓	Distillation, re-parameterizes $\text{ref}$ and $\theta$
<b>single model (no reference), CE weight <math>\lambda</math></b>				
$\ell_{\text{CPO}}$ (Xu et al., 2024)	$\log \frac{P_\theta(y_w x)}{P_\theta(y_l x)}$	✓	×	Removes $\text{ref}$
$\ell_{\text{ORPO}}$ (Hong et al., 2024)	$\log \frac{P_\theta(y_w x)(1-P_\theta(y_l x))}{P_\theta(y_l x)(1-P_\theta(y_w x))}$	✓	✓	$\beta = 1$ , main loss weighted by $\alpha$ , $\lambda = 1$
$\ell_{\text{SimPO}}$ (Meng et al., 2024)	$\log \frac{P_\theta(y_w x)}{P_\theta(y_l x)} - \gamma$	×	✓	Added margin term $\gamma$ , re-formalized in Table 2

Table 6. Details of the original losses from Table 2 and others (adapted from Meng et al. (2024)), all of which were originally implemented using the logistic log-loss, i.e., each  $\ell_x = -\log \sigma(\beta \rho_\theta)$ . We also include details about whether cross-entropy regularization (**CE term**) and length normalization (**length norm.**) were used (yes ✓, no ×) along with other details (**Extra details**) (e.g., extra weight terms, specific choices about  $\beta$  or cross-entropy weight  $\lambda$ ) that we either exclude or generalize in our analysis and experiments (e.g., extra loss weighting terms  $\alpha$ ). See Winata et al. (2024) for a comprehensive review and Zhao et al. (2025) for an approach further mixes DPO and SimPO.

in Table 6 most **no reference** approaches add an additional cross-entropy term, often making the full losses in these studies equal to  $\ell_{x+\text{CE}} = \ell_x + \lambda \ell_{\text{CE}}$  (with weight  $\lambda$ ). In some cases, additional terms  $\alpha$  are assumed that we abstract over in our analysis, e.g., in  $\ell_{\text{ORPO}}$  the full loss includes an additional weight term  $\alpha$  that is added to the main loss (in our experiments below,  $\alpha$  is implicitly set to 1).

## B. Compositionality constraint

**Proposition 1** (decompilation and standard semantic loss). *Under Assumption 1, not all of the losses in Table 2 can be decompiled into the standard semantic loss.*

*Proof.* Taking  $\ell_{\text{CPO}}$  as an example, the loss equation is based on the ratio  $s_\theta(y_w, y_l)$  consisting of two predictions  $P_\theta(y_w|x)$  and  $P_\theta(y_l|x)$ , which we can translate into the propositional formulas  $P_t := \mathbf{M}(x, y_w)$  and  $P_b := \mathbf{M}(x, y_l)$ , consisting of a total of two atomic propositions. Translating this to the standard semantic loss involves finding a *single*  $P$  such that  $P_w = P$  and  $P_l = \neg P$ . To see that no such  $P$  exists, we can enumerate all 16 unique Boolean functions over variables  $\mathbf{M}(x, y_w)$  and  $\mathbf{M}(x, y_l)$  (the only variables we are allowed under Assumption 1) and verify that none yield a single formula  $P$  s.t.  $\log \frac{\text{WMC}(P; \theta)}{\text{WMC}(\neg P; \theta)} = s_\theta(y_w, y_l)$ . The same argument can be applied to each of the other non-baseline losses in the table. □

Without the compositionality assumption, one can encode any  $\rho_\theta$  as a formula using additional variables and weighting

Input	SEM( $\cdot$ )
predictions	
$\mathbf{P}_M(y   x)$	$P := \mathbf{M}(x, y)$
formulas $P$	
$P_1 \cdot P_2$	$P := \mathbf{And}(P_1, P_2)$
$1 - P$	$P := \mathbf{Not}(P)$
$P_1 + P_2$	$P := \mathbf{Or}(P_1, P_2)$

Table 7. Rules for the compositional translation of loss expressions into symbolic formulas. See again example in Figure 4.

schemes, as is commonly done in standard WMC encodings (Chavira & Darwiche, 2008). However, the semantics of the resulting formulas are less transparent and often hidden in the weights. We instead propose to define below a novel (unweighted) encoding for preference that doesn’t require additional variables, thereby facilitating a compositional and transparent translation from loss equations.

## C. Semantic translation rules

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

## D. Proofs of propositions

Below we state propositions discussed in Section 5.1 with their proofs.

**Proposition 3** (monotonicity). *If  $\bar{P}^{(1)} \sqsubseteq \bar{P}^{(2)}$  then  $\ell_{sl}(\bar{P}^{(1)}, \theta, D) \geq \ell_{sl}(\bar{P}^{(2)}, \theta, D)$  for any  $\theta, D$ .*



$\mathbf{M}(\mathbf{x}, \mathbf{y}_w)$	$\mathbf{M}(\mathbf{x}, \mathbf{y}_l)$	$\ell_{\text{ORPO}}$	$\ell_{\text{cUnl}}$	$\ell_{13}$	$\ell_{\text{CEUnl}}$	$\ell_{\text{CCPO}}$	$\ell_{\text{CPO}}$	$\ell_{\text{CE}}$	$\ell_{\text{sCE}}$
T	T		X		X	✓	✓ X	✓	✓ X
T	F	✓	✓	✓	✓	✓	✓	✓	✓
F	T	X	X	X	X	X	X	X	X
F	F			X	X			X	X

$\mathbf{M}(\mathbf{x}, \mathbf{y}_w)$	$\mathbf{M}(\mathbf{x}, \mathbf{y}_l)$	$\ell_{\text{cfUnl}}$	$\ell_{\text{fUnl}}$	$\ell_{\text{qfUnl}}$	$\ell_{120}$	$\ell_{\text{uncPO}}$	$\ell_{114}$	$\ell_{\text{bCE}}$	$\ell_{15}$
T	T		X		X	✓	✓ X	✓	✓ X
T	F	✓	✓	✓	✓	✓	✓	✓	✓
F	T	X	X	X	X	X	X	X	X
F	F	✓	✓	✓ X	✓ X	✓	✓	✓ X	✓ X

Figure 7. A Boolean representation (in the style of Figure 3) of the single model loss functions shown in Figure 5. See again Figure 3 for how to interpret the corresponding losses.

*Proof.* By the definition of preference entailment, we have  $\bar{\mathbf{P}}_f^{(1)} \models \bar{\mathbf{P}}_f^{(2)}$ . This means that for any  $d$ ,  $\bar{\mathbf{P}}^1(d) \models \bar{\mathbf{P}}^2(d)$ , which implies that for any  $\theta$ ,  $\text{WMC}(\bar{\mathbf{P}}^{(1)}(d); \theta) \leq \text{WMC}(\bar{\mathbf{P}}^{(2)}(d); \theta)$ . From the definition of preference entailment, we also have  $\neg \bar{\mathbf{P}}^{(2)}(d) \models \neg \bar{\mathbf{P}}^{(1)}(d)$ . Following a similar line of reasoning as above, this implies  $\text{WMC}(\neg \bar{\mathbf{P}}^{(1)}(d); \theta) \geq \text{WMC}(\neg \bar{\mathbf{P}}^{(2)}(d); \theta)$ . Thus, for any  $d$  and  $\theta$ , the weighted model counting ratio term in the semantic loss in Table 3 is no larger for  $\bar{\mathbf{P}}^{(1)}$  than for  $\bar{\mathbf{P}}^{(2)}$ . It follows that  $\ell_{\text{sl}}(\bar{\mathbf{P}}^{(1)}, \theta, \{d\}) \geq \ell_{\text{sl}}(\bar{\mathbf{P}}^{(2)}, \theta, \{d\})$ . Taking the expectation over  $d \sim D$ , we obtain  $\ell_{\text{sl}}(\bar{\mathbf{P}}^{(1)}, \theta, D) \geq \ell_{\text{sl}}(\bar{\mathbf{P}}^{(2)}, \theta, D)$ .  $\square$

It follows that equivalent preference structures have identical semantic losses:

**Corollary 1** (semantic equivalence). *If  $\bar{\mathbf{P}}^1 \equiv \bar{\mathbf{P}}^2$  then  $\ell_{\text{sl}}(\bar{\mathbf{P}}^{(1)}, \theta, D) = \ell_{\text{sl}}(\bar{\mathbf{P}}^{(2)}, \theta, D)$  for any  $\theta, D$ .*

The next result is an analogue to the locality property in the original semantic loss (Xu et al., 2018), which tells us that unused logical variables in formulas do not affect loss values, which allows us to compare losses with different number of variables.

**Proposition 6** (locality). *Let  $\bar{\mathbf{P}}$  be a preference structure defined over probabilistic prediction variables  $\mathbf{X}$  with parameters  $\theta_x$ . Let  $\mathbf{Y}$  be some disjoint set of variables with parameters  $\theta_y$ . Then  $\ell_{\text{sl}}(\bar{\mathbf{P}}, \theta_x, D) = \ell_{\text{sl}}(\bar{\mathbf{P}}, [\theta_x \theta_y], D)$  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 the standard semantic loss, the probability of the world  $\mathbf{w}_{x,y}$  in the  $(\mathbf{X}, \mathbf{Y})$  space can be written as  $P_{\theta_x, \theta_y}(\mathbf{w}_{x,y}) = \prod_{X_i \in \mathbf{X}} Q_{\theta_x, \theta_y}(X_i) \cdot \prod_{Y_j \in \mathbf{Y}} Q_{\theta_x, \theta_y}(Y_j)$  where  $Q$  is either  $P$  or  $1 - P$ . Since the parameters  $\theta_x$  and

$\theta_y$  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  $P_{\theta_x, \theta_y}(\mathbf{w}_x) = \sum_{\mathbf{Y}} \left( \prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) \cdot \prod_{Y_j \in \mathbf{Y}} Q_{\theta_y}(Y_j) \right) = \prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) \cdot \sum_{\mathbf{Y}} \left( \prod_{Y_j \in \mathbf{Y}} Q_{\theta_y}(Y_j) \right) = \prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) \cdot \prod_{Y_j \in \mathbf{Y}} (Q_{\theta_y}(Y_j) + (1 - Q_{\theta_y}(Y_j))) = \prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) = P_{\theta_x}(\mathbf{w}_x)$ . This last expression is precisely the probability of the world  $\mathbf{w}_x$  in only the  $\mathbf{X}$  space. Thus,  $P_{\theta_x}(\mathbf{w}_x) = P_{\theta_x, \theta_y}(\mathbf{w}_x)$ , which implies  $\text{WMC}(\bar{\mathbf{P}}; \theta_x) = \text{WMC}(\bar{\mathbf{P}}; \theta_x, \theta_y)$  and similarly for  $\neg \bar{\mathbf{P}}$ . From this, the claim follows immediately.  $\square$

## E. New losses in loss lattice

To visualize the semantics of the single model losses shown in Figure 5, we use the Boolean truth table shown in Figure 7. As already illustrated in Figure 3, each loss column can be mechanically converted into a preference structure via the following steps: 1) translate  $\checkmark$  and  $\times$  into two standard propositional formulas that are logically consistent with the marks,  $P_t$  for  $P_b$ , respectively, then 2) apply the rules in Algorithm 1 to these formulas to get a preference structure  $\bar{\mathbf{P}}$ . (Note that the formulas in boxes in Figure 5 show the core formula  $P$  in the resulting preference structure and intentionally hide details about the constraints.)

With these preference structures, we can then obtain a compiled version of the loss by simply applying one of the versions of the semantic loss. In simplified terms, finding the compiled loss equation directly from a truth table for a given version of semantic loss with convex function  $f$  (e.g., those listed in Table 3) involves the following

$$f\left(\log \frac{\sum \checkmark}{\sum \times}\right)$$

$(M) \text{Ref}(x, y_w)$	$M(x, y_l)$	$(M) \text{Ref}(x, y_l)$	$M(x, y_w)$	$\ell_{\text{DPO/SimPO}}$	$\ell_{\text{ORPO-ref}}$	$\ell_{\text{qFUNL-ref}}$	$\ell_{15\text{-ref}}$
F	F	F	F				✓
F	F	F	T				✓
F	F	T	F			✓	✓
F	F	T	T	✓	✓	✓	✓
F	T	F	F				
F	T	F	T				✓
F	T	T	F				
F	T	T	T	✓			✓
T	F	F	F			✗	✓ ✗
T	F	F	T				✓
T	F	T	F			✓ ✗	✓ ✗
T	F	T	T	✓	✓	✓	✓
T	T	F	F	✗	✗	✗	✗
T	T	F	T	✗			✓ ✗
T	T	T	F	✗	✗	✗	✗
T	T	T	T	✓ ✗			✓ ✗

Figure 8. Boolean semantics of DPO and SimPO (column 5) and some novel variants of (columns 6-8) representing the different semantic regions in Figure 6.

where we can replace each  $\sum$  with the corresponding WMC equations for each mark, then simplify the resulting equation (i.e., the core loss equation) to arrive at a compact loss equation that can be directly used for implementation.

**Losses used in experiments** Employing the process above, below we show the core loss equations for the losses we used in our experiments in accordance with the form in Table 2:

Loss name	Core loss equation (implementation)
$\ell_{\text{CPO}}$	$\log \frac{P_\theta(y_w x)}{P_\theta(y_l x)}$
$\ell_{\text{ORPO}}$	$\log \frac{P_\theta(y_w x)(1-p_\theta(y_l x))}{P_\theta(y_l x)(1-p_\theta(y_w x))}$
$\ell_{\text{CCPO}}$	$\log \frac{P_\theta(y_w x)}{(1-P_\theta(y_w x))P_\theta(y_l x)}$
$\ell_{\text{qFUNL}}$	$\log \frac{(1-P_\theta(y_l x))}{(1-P_\theta(y_w x))}$
$\ell_{\text{cFUNL}}$	$\log \frac{(1-P_\theta(y_l x))}{(1-P_\theta(y_l x))P_\theta(y_l x)}$
$\ell_{\text{unCPO}}$	$\log \frac{P_\theta(y_l x)P_\theta(y_w x)+(1-P_\theta(y_l x))}{P_\theta(y_l x)(1-P_\theta(y_w x))}$

As described above, the final loss that we implemented was then obtained by applying the logistic loss over these equations and adding a  $\beta$  term and cross-entropy terms (see details below). We used the `trl` library for implementation from (von Werra et al., 2020), with assistance from the trainer scripts used in Meng et al. (2024).<sup>5</sup>

**Extending the loss lattice to reference models** While our loss lattice and the subsequent experiments we describe center around novel no reference loss functions, we note that given abstract structure of DPA, we can easily transform a no

<sup>5</sup>see <https://github.com/huggingface/trl> and <https://github.com/princeton-nlp/SimPO>.

reference loss function into reference loss function by simply subtracting the reference log win-lose ratio,  $s_{\text{ref}}(y_w, y_l)$  (either using a real reference ratio or one for simpo) from any single model loss equation (e.g., any of the loss equations above). Via some algebraic simplification, we can then arrive a new core loss equation with this reference information and straightforwardly generate a preference structure via Algorithm 1.

Figure 6 shows the result of this process for the single loss functions derived in Figure 5. This reveals a wide range of novel variants of DPO that we leave for future experiments and study. Figure 8 shows the Boolean semantics of DPO/SimPO and some novel variants based on the reference form of ORPO ( $\ell_{\text{ORPO-ref}}$ ), qFUNL ( $\ell_{\text{qFUNL-ref}}$ ) and 15 ( $\ell_{15\text{-ref}}$ ).

**Computing preference structures** Figure 9 shows how to symbolically compute preference structure representations in Python using the computer algebra library Sympy (Meurer et al., 2017). Specifically, lines 8-12 show how to compute a preference structure in the no-reference case, and lines 14-20 show how to compute a reference form of ORPO by adding a reference ratio.

## F. Experiments and Case studies

Our formal analysis reveals that the space of DPA losses is large, yet structured in systematic ways that we can now describe through symbolic encodings. Through case studies involving the new losses in Figure 5, we discuss some empirical results that give tips for how to better navigate this space and look for improved DPA losses using our framework.

```

1 from sympy import *
2 # winner (W), loser (L),
3 # (ref) winner (R_w), loser (R_l)
4 W,L,R_w,R_l = symbols('W,L,R_w,R_l')
5 ## equation translation for ORPO
6 P_t = And(W,Not(L))
7 P_b = And(L,Not(W))
8 ## pref. structure  $\bar{P} = (P, P_C, P_A)$ 
9 P = Implies(P_b,P_t).simplify()
10 assert P.equals(Implies(L,W))
11 P_C = Or(P_t,P_b).simplify()
12 P_A = And(P_t,P_b).simplify()
13 ## The reference form formula
14 P_ref = Implies(
15     And(P_b,R_w), And(P_t,R_l)
16 ).simplify()
17 assert P_ref.equals(
18     Implies(And(R_w,L),W)
19 )
    
```

Figure 9. An example showing how to compute the simplified symbolic formulas in preference structures for ORPO (see Figure 4) in Sympy (Meurer et al., 2017).

Specifically, we focus on losses around the known loss  $\ell_{\text{CPO}}$ , which we treat as a natural baseline to compare against. All experiments are performed using a 0.5 billion parameter LLM, Qwen-0.5B (Bai et al., 2023), tuned using trl (von Werra et al., 2020) on the ultrafeedback dataset; following standard practice, losses were implemented with a weighted cross-entropy regularizer term.

While these experiments are small scale and limited in scope, they are merely meant to suggest possible uses our framework and open questions. We also share some general observations and conjectures that we hope motivates future research in this area.

Below we provide details of the experiment setting then discuss some results and observations.

**Dataset and Model** Following much of the DPA work we cite, we train models on the ultrafeedback dataset (Cui et al., 2023), which contains around 60k binarized preference pairs aggregated from several individual preference datasets (the different categories are listed in Table 5). For tuning (detailed below) we used a custom held-out development set containing around 1.3k examples taken from the train set and reserve the test set (containing 2k examples) for final evaluation.

Standardly, we ran experiments starting from a instruction

tuned model (SFT), using a Qwen-0.5B (containing .5 billion parameters) base model (Bai et al., 2023) that was initially tuned on 6k pairs from the deita dataset of (Liu et al., 2023). To avoid repeating the process of instruction tuning, we started from the trained Qwen model released in the TRL library<sup>6</sup>.

**Hyper-parameters and model selection** The following are the standard set of tunable hyper-parameters involved in our experiments: the  $\beta$  term for DPA losses (see again Table 1), the learning rate, number of epochs, batch size and length normalization. Following other studies, we also regularized our losses with cross-entropy terms (CE) that include a tunable weight parameter  $\lambda$  that controls their contribution to the gradient. Specifically, we kept set  $\beta$  to 1, and experimented with learning rates in the range  $\{1e-6, 3e-6, 8e-6, 9e-7\}$ , number of epochs in the range of  $\{3, 5, 8\}$  and batches sizes in the range  $\{32, 128\}$  (for efficiency reasons, most tuning with done with a batch size of 32), which follow many of the suggested ranges in Meng et al. (2024). Importantly, length normalization was used throughout to make all losses comparable and given that it has been shown to improve training performance (Meng et al., 2024). We used  $\lambda$ s in the range of  $\{0.0, 0.01, 0.1, 0.3, 1.0\}$  (we found lower values, around 0.01 and 0.1, to be most effective).

For each loss function we searched the best hyper-parameters by performing a comprehensive grid search over the ranges detailed above. Final model selection was then performed by performing inference with each trained model on our held-out development set and scoring the resulting generating outputs using an off-the-shelf reward model, in particular, a 1.8B parameter reward model from (Cai et al., 2024)<sup>7</sup>. We then selected the models with the highest average reward score over the development set for comparison.

For the log probability experiments shown in Figure 10, we kept the learning rate, epoch and cross-entropy term constant (with learning rate equal to  $1e-6$ , 3 epochs, and a low cross-entropy term 0.01) to directly compare the different approaches and try to bring out their more extreme behavior.

**Evaluation protocol and win-rate comparison** We compare models tuned using our different losses using a procedure similar to how model selection is performance, which also follows the setup in Hong et al. (2024). Specifically, we do a instance-level comparison of the reward score given for each generated output, compare that score with the score of our baseline  $\ell_{\text{CPO}}$  and compute an overall win-rate, i.e., % of instances where the reward score is higher than or equal to

<sup>6</sup><https://huggingface.co/trl-lib/qwen1.5-0.5b-sft>

<sup>7</sup>[internlm/internlm2-1\\_8b-reward](https://huggingface.co/internlm/internlm2-1_8b-reward)

the reward score for  $\ell_{\text{CPO}}$  (we consider cases where items are equal given that some tasks involve generating single token output, such as the identifier of a multiple choice question or **yes** or **no**). We report the average win-rate averaged over 3 runs of each models with different generation seeds.

### F.1. Results and discussion

**How does constrainedness relate to loss behavior? Unintentional alignment shortcuts** Moving left to the right in Figure 5 yields semantically less constrained losses. For example, we see through the Boolean semantics in Figure 10 that some unconstrained losses can be satisfied by making the winner and loser both false ( $\ell_{\text{unCPO}}$ ,  $\ell_{\text{cfUNL}}$ ) or by making the the winner and loser both true ( $\ell_{\text{unCPO}}$ ,  $\ell_{\text{cfUNL}}$ ). One natural question is: *How does constrainedness contribute to a loss functions empirical success?*

We observe, consistent with other recent work on neuro-symbolic modeling (Marconato et al., 2024; van Krieken et al., 2024b), that such unconstrainedness can yield extreme behavior as illustrated in Figure 10. For example,  $\ell_{\text{unCPO}}$  and  $\ell_{\text{cfUNL}}$  attempt to make both the winners and losers false by driving their probability in the direction of zero (as shown in in both training (b) and evaluation (c)), whereas  $\ell_{\text{cfUNL}}$  keeps both probabilities high to make both true. When viewing learning as a constraint satisfaction problem, such behavior makes sense and could help to better understand various spurious training behavior observed elsewhere in the DPA literature, e.g., related to likelihood displacement and *unintentional unalignment* studied in Razin et al. (2024) or issues with preference ranking (Chen et al., 2024a).

These results suggest that understanding the way in which a loss is constrained and whether it gives rise to spurious or **unintentional alignment shortcuts** (e.g., making both predictions false) is an important factor when designing new loss functions. We note that existing losses in Figure 5 are in the middle of the two extreme points and seem less susceptible to such extreme behavior, which could explain their success.

**Can we find empirically improved losses using our method? Formalize and refine** Our ultimate aim to use our framework to help discover new and successful preference algorithms. Given the spurious behavior of losses  $\ell_{\text{unCPO}}$  and  $\ell_{\text{cfUNL}}$ , we would expect them to be less empirically successful. To test this and compare against  $\ell_{\text{CPO}}$ , we performed a model-as-judge-style experiment based on (Hong et al., 2024) that uses an off-the-shelf reward model (Cai et al., 2024) to score the outputs generated by our new models using the prompts from the *ultrafeedback* test set. We then compare these rewards scores against those of  $\ell_{\text{CPO}}$  to compute a win-rate, which gives an indication of improved or comparable generation quality over  $\ell_{\text{CPO}}$ . In-

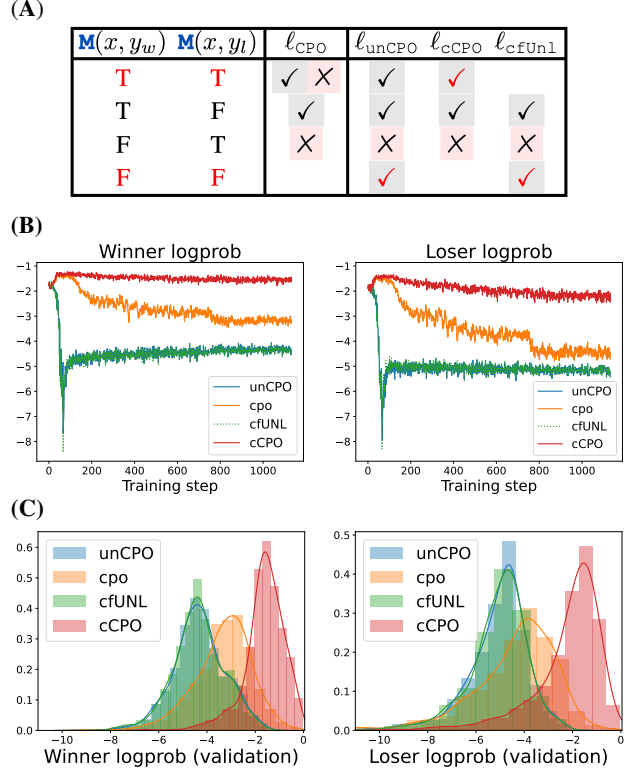


Figure 10. An illustration (A) of how to semantically satisfy losses (✓) and the corresponding log probability behavior during training (B) and evaluation (C).

deed, we see in Table 5 that in aggregate,  $\ell_{\text{unCPO}}$  and  $\ell_{\text{cfUNL}}$  have the lowest win-rate against  $\ell_{\text{CPO}}$ . Interestingly, we see that  $\ell_{\text{cCPO}}$  has a win-rate that suggests comparable generation quality to  $\ell_{\text{CPO}}$ , which shows the potential of using our framework to derive new and empirically successful losses.

These experiments are an exercise in an approach we call **formalize and refine**, i.e., starting from empirically successful losses such as  $\ell_{\text{CPO}}$ , one can formalize such losses then modify the semantics to be more or less constrained based on empirical findings. We think more large scale exploration of the full loss space, especially for DPO, is a promising direction of future research.

### Is there a single semantics for all preference learning?

**The different semantics conjecture** We note that win-rate across different categories in *ultrafeedback* (i.e., the right most columns in Table 5) varies quite considerably across models and loss types. This suggests that different types of preference data rely on a different semantics of preference, which requires a tuning approach that’s tailored to those differences. We conjecture that such a phenomenon is likely to be wide spread across different tasks and datasets, and we see more empirical work on understanding the kinds of semantics needed in different scenarios as a promising direction of future work. Such work will benefit for recent



attempts as incorporating more fine-grained annotation into preference such, such as in [Miranda et al. \(2024\)](#).

## G. DPOP equation

The DPOP loss function in Table 2 adds to the DPO an additional log term  $\alpha \cdot \max(0, \log \frac{P_{\text{ref}}(y_w | x)}{P_{\theta}(y_w | x)})$  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  $\alpha$  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

$$\rho_{\theta} = \log \frac{P_{\text{ref}}(y_l | x) P_{\theta}(y_w | x)^2}{P_{\text{ref}}(y_w | x)^2 P_{\theta}(y_l | x)}$$

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  $P_{\theta}(y_p | x)^2$  and  $P_{\text{ref}}(y_l | x)^2$  set to  $P_{\theta}(y_p | x) P_{\theta_2}(y_p | x)$  and  $P_{\text{ref}}(y_l | x) P_{\text{ref}_2}(y_l | x)$  using the copied prediction variables  $P_{\theta_2}(\cdot)$  and  $P_{\text{ref}_2}(\cdot)$ . This type of variable copying also allows us to take into account the  $\alpha$  and max above by setting the values of these copied variable to be 1 whenever the log ratio is less than 0.

Below we show the core semantic formula for DPOP, which, as noted before, makes a small adjustment to the DPO semantics as shown in Table 4:

$P := \text{Implies}(\text{And}(\text{Ref}(\mathbf{x}, y_w), \text{Ref}_2(\mathbf{x}, y_w), \mathbf{M}(\mathbf{x}, y_l)), \text{And}(\text{Ref}(\mathbf{x}, y_l), \mathbf{M}(\mathbf{x}, y_w), \mathbf{M}_1(\mathbf{x}, y_w)))$

## H. Fuzzy derivations and semantics

In contrast to the probabilistic logic approach pursued in our paper, real-valued fuzzy logics ([Zadeh, 1975](#)) extend and relax classical logic by allowing truth values to have a continuous range. In these systems, traditional Boolean operators take the form of continuous functions, based on the theory of t-norms ([Klement et al., 2013](#)), which provides a means for directly translating logic into a differentiable form. As such, they have been widely used in machine learning as a way to integrate symbolic knowledge into learning ([van Krieken et al., 2022](#); [Rocktäschel et al., 2015](#); [Donadello et al., 2017](#); [Minervini & Riedel, 2018](#); [Marra et al., 2019](#), *inter alia*).

**No reference approach** For example, in Table 8 we define the semantics of the  $\mathcal{R}$ -product variant of fuzzy logic studied in [Grespan et al. \(2021\)](#); [Giannini et al. \(2023\)](#), which we use to derive fuzzy formulas for our preference losses. For

Boolean logic	$\mathcal{R}$ -Product
<b>And</b> ( $a, b$ )	$a \cdot b$
<b>Not</b> ( $a$ )	$1 - a$
<b>Or</b> ( $a, b$ )	$a + b - a \cdot b$
<b>Implies</b> ( $a, b$ )	$\min(1, \frac{b}{a})$

Table 8. The translation of classical logic operators to  $\mathcal{R}$ -product logic for Boolean propositions  $a, b$  and their relaxed versions  $\mathbf{a}, \mathbf{b}$ .

convenience, we will use  $\text{FUZZY}(P; \theta)$  to denote the relaxed value of a formula  $P$  under the semantics in Table 8 (for simplicity, we define this fuzzy function in terms of single formulas  $P$  instead of preference structures). The fuzzy loss for  $P$  is then defined as below:

$$\ell_{\text{fuzz}}(P, \theta, D) := \mathbb{E}_{d \sim D} \left[ -\log \text{FUZZY}(P_d) \right]. \quad (7)$$

For the single model case, taking the core formulas to be  $P$  to be the following (i.e., the core semantic formula for CPO):

$P_{\text{CPO}} := \text{Implies}(\mathbf{M}(\mathbf{x}, y_l), \mathbf{M}(\mathbf{x}, y_w))$

we see the following holds (via algebraic manipulation):

$$\begin{aligned} -\log \text{FUZZY}(P_{\text{CPO}}) &= -\log \min \left( 1, \frac{P_{\theta}(y_w | x)}{P_{\theta}(y_l | x)} \right) \\ &= \max \left( 0, -\log \frac{P_{\theta}(y_w | x)}{P_{\theta}(y_l | x)} \right) \end{aligned}$$

Making the  $\ell_{\text{fuzz}}(P_{\text{CPO}}, \theta, D)$  equal to perceptron-style loss RRHF ([Yuan et al., 2023](#)) in Table 1. Given that the same core semantic formula above can be recovered between the fuzzy and probabilistic approaches, we see this as giving additional motivation to using our preference structure representations.

**DPO and reference approaches** For DPO we see a similar derivation. Given the same core formula from Table 4:

$P_{\text{DPO}} := \text{Implies}(\text{And}(\text{Ref}(\mathbf{x}, y_w), \mathbf{M}(\mathbf{x}, y_l)), \text{And}(\text{Ref}(\mathbf{x}, y_l), \mathbf{M}(\mathbf{x}, y_w)))$

we see the following explicit derivation into fuzzy logic:

$$\text{FUZZY}(P_{\text{DPO}}) = \min \left( 1, \frac{P_{\text{ref}}(y_l | x) P_{\theta}(y_w | x)}{P_{\text{ref}}(y_w | x) P_{\theta}(y_l | x)} \right)$$

and  $-\log \text{FUZZY}(P_{\text{DPO}})$  equal to:

$$\max \left( 0, -\left( \log \frac{P_{\theta}(y_w | x)}{P_{\text{ref}}(y_w | x)} - \log \frac{P_{\theta}(y_l | x)}{P_{\text{ref}}(y_l | x)} \right) \right)$$

yielding once again a perceptron-style version of DPO  $\ell_{\text{fuzz}}(P_{\text{DPO}}, \theta, D)$  similar to the RRHF approach.

**Derivation for Fuzzy Logic** To perform decompilation with fuzzy logic, we can employ a variant Algorithm 1 that removes lines 4 and 5 and makes  $P_C := \top$  and  $P_A := \perp$  by default. Importantly, given the syntactic nature of fuzzy logic, whether or not SIMPLIFY is applied in line 3 will give rise to different fuzzy loss values since the fuzzy loss is not invariant to logical equivalence (see discussion in [Marra et al. \(2024\)](#)).