
Declarative Characterizations of Direct Preference Alignment Algorithms

Anonymous Author(s)

Affiliation

Address

email

Abstract

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

19 1 Introduction

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

29
30
31
32
33 We specifically investigate direct prefer-
34 ence learning algorithms, such as direct
35 preference optimization (DPO) (Rafailov
36 et al., 2024), for pairwise preference learn-

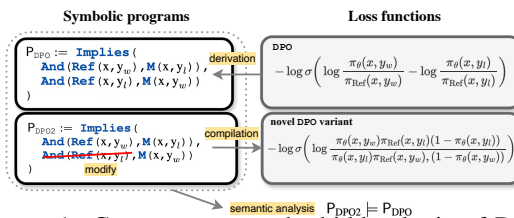


Figure 1: Can we uncover the hidden logic of DPO? Here we show the distillation of the DPO loss to a symbolic expression that expresses its high-level model behavior, along with a modified version of that program that we can compile into a novel DPO loss.

37 ing, which are currently at the forefront of research on LLM alignment and learning from human
38 preferences (Ouyang et al., 2022; Wang et al., 2023). While there has been much recent work on
39 algorithmic variations of DPO (Azar et al., 2023; Hong et al., 2024; Meng et al., 2024, *inter alia*)
40 that modify or add new terms to the original loss, understanding the differences between these new
41 proposals, as well as coming up with new variants, remains a formidable challenge due to the lack
42 of a conceptual and technical framework for reasoning about their underlying semantics.

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

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

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

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

76 2 Related work

77 **Language model alignment** While traditional approaches to language model alignment have em-
78 ployed reinforcement learning (Ziegler et al., 2019; Christiano et al., 2017), we focus on DPA ap-
79 proaches such as DPO (Rafailov et al., 2024) and SLiC (Zhao et al., 2023) that use closed-form loss
80 functions to tune models directly to offline preferences.

81 We touch on two recent areas in this space: formal characterizations of DPA losses (Azar et al.,
82 2023; Tang et al., 2024; Hu et al., 2024) and work on devising algorithmically enhanced variants
83 of DPO (Amini et al., 2024; Hong et al., 2024; Meng et al., 2024; Pal et al., 2024; Xu et al., 2024;
84 Ethayarajh et al., 2024; Park et al., 2024). In contrast to this work on formal characterization, which
85 focuses on the optimization properties of DPA losses and particular parameterizations like Bradley-
86 Terry, we attempt to formally characterize the semantic relationships between these variants of DPO
87 in an optimization agnostic way to better understand the structure of the DPA loss landscape.

88 **Neuro-symbolic modeling** For formalization, we take inspiration from work on compiling symbolic
89 formulas into novel loss functions (Li et al., 2019; Fischer et al., 2019; Marra et al., 2019; Asai &

90 Hajishirzi, 2020, *inter alia*), which is used for incorporating background constraints into learning
 91 that have shown to improve training robustness and model consistency. In particular, we focus on
 92 approaches based on probabilistic logic (Xu et al., 2018; Manhaeve et al., 2018; Ahmed et al., 2022,
 93 2023; van Krieken et al., 2024b).

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

100 **Language model programming** Finally, we take inspiration from recent work on formalizing LLM
 101 algorithms in terms of programming language concepts (Dohan et al., 2022; Beurer-Kellner et al.,
 102 2023; Khattab et al., 2023), with our approach being declarative in style. As such, our study relates
 103 to work on declarative programming techniques for ML (Eisner et al., 2004; De Raedt et al., 2007;
 104 De Raedt & Kimmig, 2015; Li et al., 2023; Vieira et al., 2017; van Krieken et al., 2024a).

105 3 Direct Preference Alignment

106 In this section, we review the basics of offline preference alignment, which can be defined as the
 107 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
 108 x and two possible generation outputs (often ones rated by humans), a preferred output y_w (the
 109 *winner* w) and a dispreferred output y_l (the *loser* l), the goal is to optimize a policy model (e.g., an
 110 LLM) $\pi_\theta(\cdot | x)$ to such preferences.

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

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

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

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

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

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

135 Single model approaches have the advantage of avoiding the overhead associated with having an ad-
 136 ditional reference model and can sometimes yield competitive performance when compared against

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

137 two model approaches. In the absence of a reference model, these losses are usually regularized
 138 using an added cross-entropy term, which we exclude from our current analysis.²

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

146 Accordingly, we will write $P_m(y | x)$ in
 147 place of $\pi_\theta(y | x)$ to denote the proba-
 148 bility assigned by a model m to an out-
 149 put y in a way that is agnostic to whether
 150 length normalization is used. In Table 2,
 151 we show different variants of DPO that
 152 we consider and two common baselines,
 153 the cross-entropy loss ℓ_{CE} and a variant
 154 that uses an unlikelihood (Welleck et al.,
 155 2019) term ℓ_{CEUnl} . Importantly, we later
 156 express each ρ_θ as a single log ratio
 157 $\rho_\theta^t / \rho_\theta^b$, which we refer to as the **core loss**
 158 **equation** for each individual loss.

159 To more easily see the relationships be-
 160 tween these proposals, we rewrite each
 161 ρ_θ in terms of the log ratio function
 162 $s_m(y_1, y_2)$ defined in Table 2 (we use
 163 \bar{y} to denote the negation of y , or $1 -$
 164 $P_m(y | x)$). Here we see that all losses
 165 are derivable from the log ratio of win-
 166 ner and loser $s_\theta(y_w, y_l)$ used in S1iC
 167 and RRHF either exactly, as in CPO (Xu et al., 2024), or with added terms. DPO, for example, is ex-
 168 pressible as this ratio minus an additional log ratio term $s_{\text{ref}}(y_w, y_l)$ that contains information about
 169 the reference model. Many variations to DPO then involve making the following two modifications.

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

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

179 4 Preference modeling as a reasoning problem

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

183 **Model predictions are symbolic objects** The declarative approach will involve thinking of LLMs
 184 predictions as logical propositions. For example, when a model M generates an output y_w for a

Loss	$\rho_\theta := \log \frac{\rho_\theta^t}{\rho_\theta^b}$	$s_{m_1, m_2}(y_1, y_2) := \log \frac{P_{m_1}(y_1 x)}{P_{m_2}(y_2 x)}$
Baselines ρ_θ		
ℓ_{CE}	$\log \frac{P_\theta(y_w x)}{1 - P_\theta(y_w x)}$	$\ell_{\text{CEUnl}} \log \frac{P_\theta(y_w x)(1 - P_\theta(y_l x))}{P_\theta(y_l x) + (1 - P_\theta(y_w x))}$
Single model approaches (no reference) P_θ		
ℓ_{CPO}	$\log \frac{P_\theta(y_w x)}{P_\theta(y_l x)}$	$s_\theta(y_w, y_l)$
ℓ_{ORPO}	$\log \frac{P_\theta(y_w x)(1 - P_\theta(y_l x))}{P_\theta(y_l x)(1 - P_\theta(y_w x))}$	$s_\theta(y_w, y_l) - s_\theta(\bar{y}_w, \bar{y}_l)$
ℓ_{SimPO}	$\log \frac{P_\theta(y_w x)P_{\text{mref}}(y_l x)}{P_{\text{mref}}(y_w x)P_\theta(y_l x)}$	$s_\theta(y_w, y_l) - s_{\text{mref}}(y_w, y_l)$
with reference model P_{ref}		
ℓ_{DPO}	$\log \frac{P_\theta(y_w x)P_{\text{ref}}(y_l x)}{P_{\text{ref}}(y_w x)P_\theta(y_l x)}$	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l)$
ℓ_{DPOP}	$\log \frac{P_\theta(y_w x)P_{\theta 2}(y_w x)P_{\text{ref}}(y_l x)}{P_{\text{ref}}(y_w x)P_{\text{ref}2}(y_w x)P_\theta(y_l x)}$	$s_\theta(y_w, y_l) - s_{\text{ref}}(y_w, y_l) - s_{\text{ref}2, \theta 2}(y_w, y_w)$

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

²When referring to the CPO, ORPO and S1iC losses, we refer to the losses without the cross-entropy terms. For example, what we call S1iC and ODPO refers to the ca1 and OR losses, respectively, in the original papers.

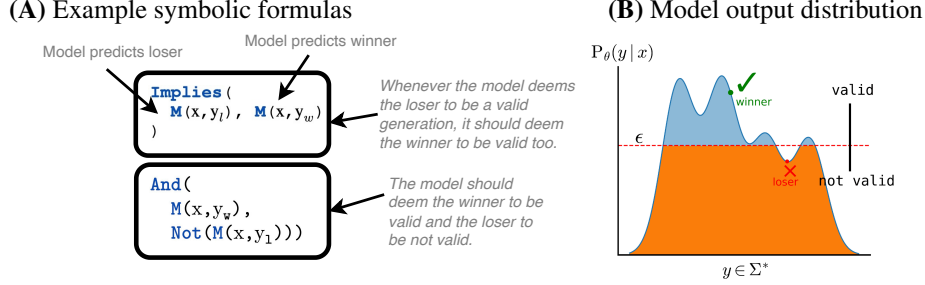


Figure 2: What do formal representations of loss functions tell us? We show (A) two symbolic formulas related to single model preference learning with their semantics paraphrased in informal English. When grounded in model behavior, they tell us about the structure of the model’s output probability distribution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

185 prompt x , we will use the notation $M(x, y_w)$ to express the proposition that y_w is a valid generation
 186 for x . Importantly, we will further weight these propositions by assigning the probabilities given
 187 by the corresponding model, i.e., $P_\theta(M(x, y_w)) = P_\theta(y_w | x)$. We call these propositions our
 188 **probabilistic predictions** X_1, \dots, X_n , which will form the basis of symbolic formulas.

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

196 When grounded to model behavior via the proposition weights, such constraints tell us about the
 197 structure of a model’s output probability distribution, as visualized in Figure 2 (B). Semantically, we
 198 assume that what constitutes a valid generation is any probabilistic prediction whose weight exceeds
 199 some threshold ϵ in that distribution, similar to the notion of ϵ -truncated support from Hewitt et al.
 200 (2020). While our results later will not depend on making any direct assumptions about ϵ , such a
 201 definition is merely meant to provide intuitions for how to understand our formulas.

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

206 4.1 Compilation and Decompileation

207 **Compilation and semantic loss** To compile a symbolic formula P into loss, we employ a proba-
 208 bilistic approach based on the semantics of weighted model counting (WMC) (Chavira & Darwiche,
 209 2008; Fierens et al., 2015). This is based on computing a probability of a formula P , given by

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

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

215 Formally, the semantic loss takes the form $\mathbb{E}_{d \sim D}[-\log p_\theta(P_d)]$, where we use the notation P_d to
 216 refer to the substitution of variables in our formulas P (e.g., x, y_w, y_l) with specific values from $d \sim$

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

217 *D.* Since our approach will later involve computing the probability of P conditioned (optionally)
 218 on some **conditioning constraints** P_C (i.e., an additional propositional formula), we consider the
 219 conditional form of the semantic loss and its full objective below:

$$\min_{\theta} \mathbb{E}_{d \sim D} \left[-\log p_{\theta}(P_d | P_{C_d}) \right], \quad p_{\theta}(P | P_C) = \frac{\text{WMC}(P \wedge P_C; \theta)}{\text{WMC}(P \wedge P_C; \theta) + \text{WMC}(\neg P \wedge P_C; \theta)} \quad (3)$$

220 where the last part follows from the standard definition of conditional probability (with the denomi-
 221 nator being an expanded form of $\text{WMC}(P_C; \theta)$). We note that when P_C is equal to \top (or true), this
 222 form of the semantic loss is equivalent to the original version.

223 As an important technical point, we see below how having an explicit negation $\neg P$ in the normaliza-
 224 tion allows us write the probability of P in the following way (without loss of generality, we exclude
 225 P_C to improve readability and remove θ from WMC):

$$p_{\theta}(P) = \frac{\exp(\log \text{WMC}(P))}{\exp(\log \text{WMC}(P)) + \exp(\log \text{WMC}(\neg P))} = \sigma \left(\underbrace{\log \frac{\text{WMC}(P)}{\text{WMC}(\neg P)}}_{\text{semantic loss ratio } \rho_{\text{sem}}} \right) \quad (4)$$

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

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

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

232 5 A logic for preference modeling

233 In the standard semantic loss (SL), ML loss functions ℓ_x are expressible as a single propositional for-
 234 mulas P interpreted via probabilistic logic, with $\ell_x \sim -\log p_{\theta}(P)$. At first glance, this formulation
 235 is at odds with standard formulations of pairwise preference, such as the Bradley-Terry (BT) model
 236 (Bradley & Terry, 1952) typically assumed in RLHF, which involves modeling a preference distri-
 237 bution $p_{\theta}(y_w \succ y_l)$ between two (often disparate) quantities (e.g., given by the kinds of log ratios
 238 in Table 2). Indeed, logical accounts of pairwise preference such as Rescher (1967) (see Cvetković
 239 (1993)) assume a similar semantics where preference is defined not as a single propositional formula
 240 but as an inequality between the counts of two independent formulas $\text{WMC}(P_w) > \text{WMC}(P_l)$.

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

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

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

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

$$\overline{P}_f := (P \vee P_A) \wedge P_C, \quad \overline{\neg P}_f := (\neg P \vee P_A) \wedge P_C. \quad (6)$$

256 In the absence of the additive constraint P_A , we note that these representations encode the condi-
 257 tional $P \mid P_C$, thus making the semantic loss of these formulas equivalent to the conditional semantic
 258 loss in Eq 3. Indeed, many DPA losses will be reducible to the conditional semantic loss, however,
 259 P_A and the ability to add default model counts to P and $\neg P$ will be needed to derive some DPA
 260 losses symbolically and account for peculiar properties of their normalization.

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

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

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

$$P_1 \equiv \left(\underbrace{(P_2 \rightarrow P_1)}_P \vee \underbrace{(P_1 \wedge P_2)}_{P_A} \right) \wedge \underbrace{(P_1 \vee P_2)}_{P_C}, P_2 \equiv \left(\underbrace{\neg(P_2 \rightarrow P_1)}_{\neg P} \vee \underbrace{(P_1 \wedge P_2)}_{P_A} \right) \wedge \underbrace{(P_1 \vee P_2)}_{P_C}$$

267 As noted above, this construction corresponds exactly to the preference structure (P, P_C, P_A) with
 268 $P := P_2 \rightarrow P_1$, $P_C := P_1 \vee P_2$ and $P_A := P_1 \wedge P_2$ and its two formula forms. (As a special
 269 case, whenever $P_2 \equiv \neg P_1$, this simplifies to the structure $\overline{P} = (P_1, \top, \perp)$, thus making any single
 270 formula representable as a preference structure.) \square

271 5.1 Generalized semantic loss based on preference structures

272 In our generalization of the semantic loss, for-
 273 mulas P will be replaced with preference struc-
 274 tures \overline{P} . For example, we can modify the logis-
 275 tic log form of SL in Eq 5 to be $\ell(\overline{P}, \theta, D)$ and
 276 change the semantic loss ratio ρ_{sem} accordingly
 277 to operate over the formula forms of \overline{P} in Eq 6.
 278 By analogy to the generalized DPA in Eq 1, we
 279 can view this logistic log form as a particular
 280 instance of a **generalized semantic loss**:

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

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

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

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

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

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

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

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

301 **Proposition 2** (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 θ, D .*

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

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

309 5.2 Decompiling DPA losses into preference structures

310 The **decompilation** of a DPA loss ℓ_{DPA_x} into a symbolic form can now be stated as find-
 311 ing a preference structure \bar{P} whose semantic loss is equal to ℓ_{DPA_x} , as given in Eq 8:

$$312 \quad \forall D, \theta. \ell_{DPA_x} = \ell_{sl}(\bar{P}, D_p, \theta) \text{ s.t.} \quad (8) \quad \rho_\theta = \rho_{sem}, \text{ with } \frac{\rho_\theta^t}{\rho_\theta^b} = \frac{\text{WMC}(\bar{P}; \theta)}{\text{WMC}(\neg \bar{P}; \theta)} \quad (9)$$

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

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

Algorithm 1: DPA to logic

Input : loss expression $\rho_\theta = \frac{\rho_\theta^t}{\rho_\theta^b}$

Output: $\bar{P} = (P, P_C, P_A)$

```

1  $P_t \leftarrow \text{SEM}(\rho_\theta^t)$ 
2  $P_b \leftarrow \text{SEM}(\rho_\theta^b)$ 
3  $P \leftarrow \text{SIMPLIFY}(\text{Implies}(P_b, P_t))$ 
4  $P_C \leftarrow \text{SIMPLIFY}(\text{Or}(P_t, P_b))$ 
5  $P_A \leftarrow \text{SIMPLIFY}(\text{And}(P_t, P_b))$ 

```

320 **Characterizing the DPA equation class** By construc-
 321 tion, we will assume that all the core equations for DPA
 322 losses ρ_θ^t and ρ_θ^b are expressible as certain types of *disjoint*
 323 multilinear polynomials over binary variables $\{x_i\}_{i=1}^n$, in-
 324 tuitively polynomials whose translation via the rules in Ta-
 325 ble 7 results in valid formulas of propositional logic. Formally, such polynomials over n variables
 326 are defined as any polynomial e of the form $e = \sum_i e_i$ where (a) for all i there exists $J_i \subseteq \{1, \dots, n\}$
 327 such that $e_i = \prod_{j \in J_i} \ell_{ij}$ where ℓ_{ij} is either x_j or $(1 - x_j)$, and (b) for all i, i' , terms e_i and $e_{i'}$ are
 328 disjoint, i.e., have no common solutions (for some k , one term has x_k and the other has $1 - x_k$).

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

332 **Translation algorithm** Our translation process is shown in Algorithm 1. Given an input ρ_θ , both
 333 parts of that equation are translated into logic (**lines 1-2**) via a translation function SEM. The trans-
 334 lation is standard and its correctness can be established via induction on the rules (see the full rules
 335 in Table 7): each model prediction $P_M(\cdot)$ is mapped to a probabilistic prediction $M(\cdot)$ then: $1 - P$
 336 is mapped to negation, $P_1 \cdot P_2$ to conjunction, and $P_1 + P_2$ to disjunction. **Lines 3-5** apply the
 337 implication construction from Prop 1 to create a \bar{P} , where formulas are minimized via SIMPLIFY.

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

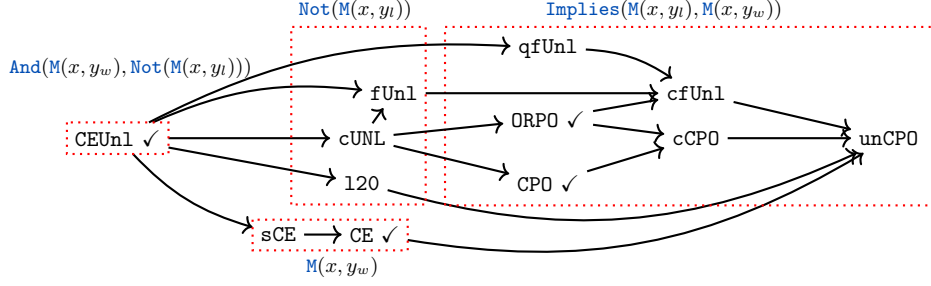


Figure 3: What other losses are there? Here we show the loss landscape for single model preference approaches using a **loss lattice** showing losses (nodes) structured according to strict entailment and their core formulas P (dashed boxes). See Appendix C for details of the individual losses.

340 **Proposition 3** (correctness). *Given a loss equation $\rho_\theta = \rho_\theta^t / \rho_\theta^b$ where ρ_θ^t , and ρ_θ^b are disjoint*
 341 *multilinear polynomials, Algorithm 1 returns a preference structure \bar{P} whose semantic loss ratio*
 342 *ρ_{sem} equals ρ_θ .*

343 6 Results and Discussion

344 Table 4 shows the preference structures obtained from Algorithm 1 for the key DPA losses in Table 2.
 345 Given that the original losses were all formulated using the logistic log form of DPA, the correctness
 346 of Algorithm 1 (Prop. 3) tells us that compiling the representations in Table 4 under ℓ_{sl-log} will yield
 347 exactly the original losses. Importantly, when the DPO symbolic form is compiled using $\ell_{sl-square}$
 348 (i.e., the squared loss form of SL), this will yield exactly IPO Azar et al. (2023), showing how our
 349 semantic analysis is invariant to the particular choice of f . (A similar argument can be made for
 350 deriving S11C from $\ell_{sl-margin}$ and the representation for CPD).

351 6.1 What we learn about existing losses?

352 **Single model approaches have an intuitive semantics, highly constrained** Under our analysis, CPD
 353 and ORPO are both derived from the same core semantic formula P and implication first introduced in Fig-
 354 ure 2, in spite of the superficial differences in their original form. They differ, however, in terms of the
 355 conditioning constraints P_C they impose, with CPD imposing a **one-true** constraint that requires either the
 356 winner or loser to be deemed valid, whereas ORPO imposes a **one-hot** constraint where one and only one can
 357 be deemed valid. When plotted in a broader loss landscape, as shown in Figure 3, we see that both are en-
 358 tailed by the CEUNL baseline, yet have a non-entailing relation to one another.
 359
 360
 361
 362
 363
 364
 365

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

375 **There are many losses still to explore** We systematically create new losses by manipulating the
 376 the conditioning constraints that existing losses impose. Figure 3 shows a (non-exhaustive) lattice

Loss	Representation \bar{P}
CE	$P := M(x, y_w), P_C := \perp$
CEUnl	$P := \text{And}(M(x, y_w), \text{Not}(M(x, y_l))),$ $P_C := \perp$
CPD	$P := \text{Implies}(M(x, y_l), M(x, y_w))$ $P_C := \text{Or}(M(x, y_l), M(x, y_w))$
ORPO	$P := \text{Implies}(M(x, y_l), M(x, y_w))$ $P_C := \text{Or}(\text{And}(M(x, y_l), \text{Not}(M(x, y_w))), \text{And}(\text{Not}(M(x, y_l)), M(x, y_w)))$
DPO	$P := \text{Implies}(\text{And}(\text{Ref}(x, y_w), M(x, y_l)), \text{And}(\text{Ref}(x, y_l), M(x, y_w)))$ $P_C := \text{Or}(\text{And}(\text{Ref}(x, y_w), M(x, y_l)), \text{And}(\text{Ref}(x, y_l), M(x, y_w)))$

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

377 representation of the loss landscape for single model preference approaches created by mechanically
378 deriving new losses from the CEUn1 baseline (i.e., the most constrained loss) and ordering them by
379 strict entailment (this terminates in unCPO, a version of CPO without conditioning; see Appendix C
380 for details). Here we see that different regions emerge characterized by different formulas P , notably
381 an entirely unexplored region between CEUn1 and CPO and ORPO of unlikelihood losses that
382 optimize for the negation of the loser ($\text{Not}(M(x, y_l))$).

383 **DPO has a peculiar semantics, shared among variants** The semantics of DPO shown in Table 4
384 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
385 $\text{Ref}(x, y_w) \wedge \neg M(x, y_l) \rightarrow \neg M(x, y_l)$. The first says that *If the reference deems the winner to be*
386 *valid and the tunable model deems the loser to be valid, then that model should also deem the*
387 *winner to be valid*, while the second says that *the tunable model should deem the loser to be not*
388 *valid whenever the reference deems the winner to be valid and the loser to be not valid*. While this
389 semantics makes sense, and complements nicely the semantics of CPO by adding information about
390 the referent model, DPO includes conditioning constraints that are hard to justify from first principles,
391 and that make it semantically disconnected from the CE and CEUn1 baselines. While DPO belongs
392 to a much larger space, we conjecture that investigating the different semantic neighborhoods that
393 result from modifying its conditioning constraints, as in Figure 3, is a promising direction.

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

399 6.2 Are any of these new losses good?

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

405 7 Conclusion

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

415 References

- 416 Kareem Ahmed, Stefano Teso, Kai-Wei Chang, Guy Van den Broeck, and Antonio Vergari. Seman-
417 tic probabilistic layers for neuro-symbolic learning. *Advances in Neural Information Processing*
418 *Systems*, 35:29944–29959, 2022.
- 419 Kareem Ahmed, Stefano Teso, Paolo Morettin, Luca Di Liello, Pierfrancesco Ardino, Jacopo Gobbi,
420 Yitao Liang, Eric Wang, Kai-Wei Chang, Andrea Passerini, et al. Semantic loss functions for
421 neuro-symbolic structured prediction. In *Compendium of Neurosymbolic Artificial Intelligence*,
422 pp. 485–505. IOS Press, 2023.
- 423 Afra Amini, Tim Vieira, and Ryan Cotterell. Direct preference optimization with an offset. *arXiv*
424 *preprint arXiv:2402.10571*, 2024.

- 425 Akari Asai and Hannaneh Hajishirzi. Logic-guided data augmentation and regularization for con-
426 sistent question answering. *ACL 2020*, 2020.
- 427 Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal
428 Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human
429 preferences. *arXiv preprint arXiv:2310.12036*, 2023.
- 430 Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. Prompting is programming: A query lan-
431 guage for large language models. *Proceedings of the ACM on Programming Languages*, 7(PLDI):
432 1946–1969, 2023.
- 433 Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method
434 of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- 435 Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Arti-
436 ficial Intelligence*, 172(6-7):772–799, 2008.
- 437 Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep
438 reinforcement learning from human preferences. *Advances in neural information processing sys-
439 tems*, 30, 2017.
- 440 Dragan Cvetković. *The Logic Preference and Decision Supporting Systems*. PhD thesis, University
441 of Beograd, 1993.
- 442 Luc De Raedt and Angelika Kimmig. Probabilistic (logic) programming concepts. *Machine Learn-
443 ing*, 100:5–47, 2015.
- 444 Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: A probabilistic prolog and its ap-
445 plication in link discovery. In *IJCAI 2007, Proceedings of the 20th international joint conference
446 on artificial intelligence*, pp. 2462–2467. IJCAI-INT JOINT CONF ARTIF INTELL, 2007.
- 447 David Dohan, Winnie Xu, Aitor Lewkowycz, Jacob Austin, David Bieber, Raphael Gontijo Lopes,
448 Yuhuai Wu, Henryk Michalewski, Rif A Saurous, Jascha Sohl-Dickstein, et al. Language model
449 cascades. *arXiv preprint arXiv:2207.10342*, 2022.
- 450 Jason Eisner, Eric Goldlust, and Noah A Smith. Dyna: A declarative language for implementing
451 dynamic programs. In *Proc. of ACL*, 2004.
- 452 Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model
453 alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.
- 454 Daan Fierens, Guy Van den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon,
455 Gerda Janssens, and Luc De Raedt. Inference and learning in probabilistic logic programs using
456 weighted boolean formulas. *Theory and Practice of Logic Programming*, 15(3):358–401, 2015.
- 457 Marc Fischer, Mislav Balunovic, Dana Drachler-Cohen, Timon Gehr, Ce Zhang, and Martin
458 Vechev. DL2: training and querying neural networks with logic. In *International Conference
459 on Machine Learning*, pp. 1931–1941. PMLR, 2019.
- 460 Dan Friedman, Alexander Wettig, and Danqi Chen. Learning transformer programs. *Advances in
461 Neural Information Processing Systems*, 36, 2024.
- 462 Francesco Giannini, Giuseppe Marra, Michelangelo Diligenti, Marco Maggini, and Marco Gori.
463 On the relation between loss functions and t-norms. In *Inductive Logic Programming: 29th
464 International Conference, ILP 2019, Plovdiv, Bulgaria, September 3–5, 2019, Proceedings 29*,
465 pp. 36–45. Springer, 2020.
- 466 Joseph Y Halpern, Robert Harper, Neil Immerman, Phokion G Kolaitis, Moshe Y Vardi, and Victor
467 Vianu. On the unusual effectiveness of logic in computer science. *Bulletin of Symbolic Logic*, 7
468 (2):213–236, 2001.
- 469 John Hewitt, Michael Hahn, Surya Ganguli, Percy Liang, and Christopher D. Manning. RNNs can
470 generate bounded hierarchical languages with optimal memory. In Bonnie Webber, Trevor Cohn,
471 Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in
472 Natural Language Processing (EMNLP)*, pp. 1978–2010, Online, November 2020. Association
473 for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.156.

- 474 Jiwoo Hong, Noah Lee, and James Thorne. Reference-free monolithic preference optimization with
475 odds ratio. *arXiv preprint arXiv:2403.07691*, 2024.
- 476 Xiangkun Hu, Tong He, and David Wipf. New desiderata for direct preference optimization. *arXiv*
477 *preprint arXiv:2407.09072*, 2024.
- 478 Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri
479 Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. Dspy:
480 Compiling declarative language model calls into self-improving pipelines. *arXiv preprint*
481 *arXiv:2310.03714*, 2023.
- 482 Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. A Logic-Driven Framework for Con-
483 sistency of Neural Models. In *Proceedings of the 2019 Conference on Empirical Methods in*
484 *Natural Language Processing and the 9th International Joint Conference on Natural Language*
485 *Processing (EMNLP-IJCNLP)*, pp. 3924–3935, Hong Kong, China, November 2019. Association
486 for Computational Linguistics. doi: 10/ghvf88.
- 487 Ziyang Li, Jiani Huang, and Mayur Naik. Scallop: A language for neurosymbolic programming.
488 *Proceedings of the ACM on Programming Languages*, 7(PLDI):1463–1487, 2023.
- 489 Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt.
490 Deepproblog: Neural probabilistic logic programming. *Advances in neural information process-*
491 *ing systems*, 31, 2018.
- 492 Giuseppe Marra, Francesco Giannini, Michelangelo Diligenti, and Marco Gori. Integrating learning
493 and reasoning with deep logic models. In *Joint European Conference on Machine Learning and*
494 *Knowledge Discovery in Databases*, pp. 517–532. Springer, 2019.
- 495 John McCarthy et al. *Programs with common sense*. RLE and MIT computation center Cambridge,
496 MA, USA, 1960.
- 497 Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a
498 reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.
- 499 Nils J Nilsson. Logic and artificial intelligence. *Artificial intelligence*, 47(1-3):31–56, 1991.
- 500 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong
501 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to fol-
502 low instructions with human feedback. *Advances in neural information processing systems*, 35:
503 27730–27744, 2022.
- 504 Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White.
505 Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint*
506 *arXiv:2402.13228*, 2024.
- 507 Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality
508 in direct preference optimization. *arXiv preprint arXiv:2403.19159*, 2024.
- 509 Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea
510 Finn. Direct preference optimization: Your language model is secretly a reward model. *Proceed-*
511 *ings of Neurips*, 2024.
- 512 Nicholas Rescher. *The logic of decision and action*. University of Pittsburgh Press, 1967.
- 513 Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Row-
514 land, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Gen-
515 eralized preference optimization: A unified approach to offline alignment. *arXiv preprint*
516 *arXiv:2402.05749*, 2024.
- 517 Emile van Krieken, Samy Badreddine, Robin Manhaeve, and Eleonora Giunchiglia. Uller: A unified
518 language for learning and reasoning. In *International Conference on Neural-Symbolic Learning*
519 *and Reasoning*, pp. 219–239. Springer, 2024a.

- 520 Emile van Krieken, Pasquale Minervini, Edoardo M Ponti, and Antonio Vergari. On the independ-
521 dence assumption in neurosymbolic learning. *arXiv preprint arXiv:2404.08458*, 2024b.
- 522 Tim Vieira, Matthew Francis-Landau, Nathaniel Wesley Filardo, Farzad Khorasani, and Jason Eis-
523 ner. Dyna: Toward a self-optimizing declarative language for machine learning applications. In
524 *Proceedings of the 1st ACM SIGPLAN International Workshop on Machine Learning and Pro-*
525 *gramming Languages*, pp. 8–17, 2017.
- 526 Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang,
527 Xin Jiang, and Qun Liu. Aligning large language models with human: A survey. *arXiv preprint*
528 *arXiv:2307.12966*, 2023.
- 529 Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston.
530 Neural text generation with unlikelihood training. In *International Conference on Learning Rep-*
531 *resentations*, 2019.
- 532 Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton
533 Murray, and Young Jin Kim. Contrastive preference optimization: Pushing the boundaries of llm
534 performance in machine translation. *arXiv preprint arXiv:2401.08417*, 2024.
- 535 Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. A Semantic Loss Function for
536 Deep Learning with Symbolic Knowledge. In *International Conference on Machine Learning*,
537 pp. 5498–5507, 2018.
- 538 Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf:
539 Rank responses to align language models with human feedback without tears. *arXiv preprint*
540 *arXiv:2304.05302*, 2023.
- 541 Yao Zhao, Mikhail Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu.
542 Calibrating sequence likelihood improves conditional language generation. In *The eleventh inter-*
543 *national conference on learning representations*, 2022.
- 544 Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. SLiC-
545 HF: Sequence Likelihood Calibration with Human Feedback. *arXiv preprint arXiv:2305.10425*,
546 2023.
- 547 Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul
548 Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv*
549 *preprint arXiv:1909.08593*, 2019.

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

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

550 A Semantic translation rules

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

552 B Proofs of propositions

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

554 **Proposition 4** (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 θ, D .*

555 *Proof.* By the definition of preference entailment, we have $\bar{P}_f^{(1)} \models \bar{P}_f^{(2)}$. This means that for any d ,
556 $\bar{P}^1(d) \models \bar{P}^2(d)$, which implies that for any θ , $\text{WMC}(\bar{P}^{(1)}(d); \theta) \leq \text{WMC}(\bar{P}^{(2)}(d); \theta)$. From the
557 definition of preference entailment, we also have $\neg \bar{P}^{(2)}(d) \models \neg \bar{P}^{(1)}(d)$. Following a similar line of
558 reasoning as above, this implies $\text{WMC}(\neg \bar{P}^{(1)}(d); \theta) \geq \text{WMC}(\neg \bar{P}^{(2)}(d); \theta)$. Thus, for any d and
559 θ , the weighted model counting ratio term in the semantic loss in Table 5 is no larger for $\bar{P}^{(1)}$ than
560 for $\bar{P}^{(2)}$. It follows that $\ell_{sl}(\bar{P}^{(1)}, \theta, \{d\}) \geq \ell_{sl}(\bar{P}^{(2)}, \theta, \{d\})$. Taking the expectation over $d \sim D$, we
561 obtain $\ell_{sl}(\bar{P}^{(1)}, \theta, D) \geq \ell_{sl}(\bar{P}^{(2)}, \theta, D)$. \square

562 **Proposition 5** (locality). *Let \bar{P} be a preference structure defined over probabilistic prediction vari-
563 ables \mathbf{X} with parameters θ_x . Let \mathbf{Y} be some disjoint set of variables with parameters θ_y . Then
564 $\ell_{sl}(\bar{P}, \theta_x, D) = \ell_{sl}(\bar{P}, [\theta_x, \theta_y], D)$ for any D .*

565 *Proof.* Let \mathbf{w}_x be any world over variables \mathbf{X} and \mathbf{w}_y be any world over (disjoint) variables \mathbf{Y} .
566 Let $\mathbf{w}_{x,y}$ denote the joint world. By Eq 2, the probability of the world $\mathbf{w}_{x,y}$ in the (\mathbf{X}, \mathbf{Y}) space
567 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
568 $1 - P$. Since the parameters θ_x and θ_y refer to disjoint sets of variables, we can simplify this to
569 $\prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) \cdot \prod_{Y_j \in \mathbf{Y}} Q_{\theta_y}(Y_j)$.

570 It follows that the marginal probability of the world \mathbf{w}_x in the (\mathbf{X}, \mathbf{Y}) space equals $P_{\theta_x, \theta_y}(\mathbf{w}_x) =$
571 $\sum_{\mathbf{Y}} \left(\prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) \cdot \prod_{Y_j \in \mathbf{Y}} Q_{\theta_y}(Y_j) \right) = \prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) \cdot \sum_{\mathbf{Y}} \left(\prod_{Y_j \in \mathbf{Y}} Q_{\theta_y}(Y_j) \right) =$
572 $\prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) \cdot \prod_{Y_j \in \mathbf{Y}} (Q_{\theta_y}(Y_j) + (1 - Q_{\theta_y}(Y_j))) = \prod_{X_i \in \mathbf{X}} Q_{\theta_x}(X_i) = P_{\theta_x}(\mathbf{w}_x)$. This last
573 expression is precisely the probability of the world \mathbf{w}_x in only the \mathbf{X} space. Thus, $P_{\theta_x}(\mathbf{w}_x) =$
574 $P_{\theta_x, \theta_y}(\mathbf{w}_x)$, which implies $\text{WMC}(\bar{P}; \theta_x) = \text{WMC}(\bar{P}; \theta_x, \theta_y)$ and similarly for $\neg \bar{P}$. From this, the
575 claim follow immediately. \square

576 C Boolean visualization of losses

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

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

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

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

585 D DPOP equation

586 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
 587 aims to ensure that the log-likelihood of preferred example is high relative to the reference model
 588 (we simplified this loss by removing the \max and α parameter, the latter of which is set to be a
 589 whole number ranging from 5 to 500 in Pal et al. (2024)). When translating the full loss into a single
 590 log, this results in the equation $\rho_{\theta} = \log \frac{P_{\text{ref}}(y_l|x)P_{\theta}(y_w|x)^2}{P_{\text{ref}}(y_w|x)^2P_{\theta}(y_l|x)}$ for $\alpha = 1$. The top and bottom equations
 591 are hence not multilinear since they both contain exponents > 1 . To fix this, we can simply create
 592 copies of these variables, e.g., with $P_{\theta}(x, y_w)^2$ and $P_{\text{ref}}(y_l | x)^2$ set to $P_{\theta}(x, y_w) P_{\theta 2}(x, y_w)$ and
 593 $P_{\text{ref}}(y_l | x) P_{\text{ref}2}(y_l | x)$ using the copied prediction variables $P_{\theta 2}(\cdot)$ and $P_{\text{ref}2}(\cdot)$.

$M(x, y_w)$	$M(x, y_t)$	Unl	qfUnl	cUnl	CE	bCE	fUnl	ORPO	cfUnl	CPO	cCPO	uncCPO
T	T	X		X	✓	✓	X			✓	✓	✓
T	F	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
F	T	X		X	X	X	X	X	X	X	X	X
F	F	X	✓		X	✓	✓		✓			✓

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