SYMRANKOR: SYMMETRIC PAIRWISE ORDINAL LEARNING FOR SEMI-SUPERVISED REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Continuous-valued labels are often costly to obtain, motivating semi-supervised regression methods that learn from abundant unlabeled data. Confidence-based classification methods do not transfer directly, because a scalar regression output provides no normalized class confidence for selecting pseudo-labels. State-of-the-art methods such as RankUp address this mismatch with an auxiliary ranking classifier (ARC) over sample pairs, but its unlabeled pairwise objective uses cross-entropy and does not explicitly account for corrupted pseudo-orders. We propose SymRankor, which treats the weak-view order prediction as a potentially corrupted pairwise pseudo-label under a Bradley–Terry interpretation and applies a symmetric loss to the strong-view score difference. Under a theoretical label-flip model, the unlabeled-pair risk is a positive affine transform of its clean counterpart. This transformation preserves risk ordering and any attained argmin set without estimating the flip rate. Experiments across vision, audio, and text show that SymRankor improves upon its direct predecessor, with the clearest gains on audio and text. The symmetric objective remains competitive across the evaluated modalities even without confidence filtering. This study explicitly examines semi-supervised regression from this perspective. A chemical case study further suggests that the same ordinal objective can exploit coarse domain-derived ordinal information.

1 INTRODUCTION

Many scientific prediction tasks involve continuous targets that are expensive to measure. Molecular-property prediction provides a natural setting for semi-supervised learning: property labels are costly to obtain, whereas unlabeled molecular structures are abundant (Ross et al., 2022). Relative judgments can be easier to obtain than absolute labels or scores (Cohen et al., 1998), making ordinal information a natural source of auxiliary learning signals.

Semi-supervised classification commonly relies on categorical confidence or prediction consistency. FixMatch generates a categorical pseudo-label from a weakly augmented view and uses it to train the model on a strongly augmented view when the maximum predicted class probability exceeds a confidence threshold (Sohn et al., 2020). We refer to these as the weak-view and the strong-view, respectively. This mechanism does not transfer directly to regression because a scalar output has no direct analogue of class-probability confidence. RankUp addresses this limitation by attaching a RankNet-style Auxiliary Ranking Classifier (ARC) to the regressor's representation (Huang et al., 2024). The ARC predicts the relative order of each pair and outputs a pairwise order probability that can be used to screen pseudo-labels. Its unlabeled pairwise term nevertheless uses binary cross-entropy, so an incorrect pseudo-order is treated in the same manner as an observed order label during optimization. This motivates shifting the focus from selective pseudo-labeling to robustness against noisy pseudo-labels.

Such robustness can instead be introduced through a symmetric loss. A margin loss is symmetric when $\ell(z) + \ell(-z) = K$ for a constant K. Under symmetric label flips, this condition makes the noisy population risk a positive affine transform of the clean risk and therefore preserves its minimizers. Charoenphakdee et al. (2019) establish related minimizer-invariance results for surrogate balanced-error-rate classification and bipartite AUC under their stated conditions. Label smoothing does not generally provide the same guarantee. Its preferred strength may decrease as label

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

noise increases (Wei et al., 2022). Classification calibration is a separate property specific to the loss, distinct from this minimizer invariance. SymPO combines preference-noise robustness with classification-calibrated reward learning to obtain rank preservation and policy improvement under its stated assumptions (Nishimori et al., 2026). Accordingly, we interpret the order inferred from the weak-view as a potentially corrupted pairwise pseudo-label and impose a symmetric loss on the strong-view score margin. This division of roles suits SymRankor because the ARC supplies ordinal regularization while the labeled regression head remains responsible for estimating target magnitudes. Confidence filtering determines which pairs enter the objective, whereas the bounded sigmoid loss limits the contribution of an incorrectly ordered pair. Because the ARC signal is ordinal, the same objective can also incorporate coarse ordinal information. For the chemical case study, we adopt the ordered-label perspective of semi-supervised ordinal regression (Tsuchiya et al., 2021). Our formulation instead retains continuous target prediction and uses the ordinal information as auxiliary learning signals.

Contributions.

- **Noise-robust unlabeled pairwise learning.** We reinterpret self-generated pairwise orders as potentially corrupted labels and replace cross-entropy on the unlabeled ARC term with a symmetric margin loss.
- **Reduced dependence on confidence filtering.** SymRankor permits training with $\tau = 0$, so a confidence gate does not limit the usable amount of unlabeled data. Semi-supervised learning documents this quantity–quality trade-off (Chen et al., 2023). Selecting unlabeled pairs by a confidence threshold therefore remains a component with room for further improvement.
- **Coarse ordinal supervision for scientific regression.** The same auxiliary objective can exploit domain-derived ordinal classes. Their potential is illustrated as pseudo-ordinal annotation.

2 Related Work

Weak supervision. Following Sugiyama et al. (2022), weak supervision studies how to learn useful predictive models from supervision that is weaker than the accurate and sufficiently abundant labels assumed by fully supervised learning. More broadly, it is learning useful predictive models from limited, coarse, or noisy supervision. Zhou (2018) distinguishes three forms of weak supervision. First, incomplete supervision is where only part of the training data is labeled. Second, inexact supervision refers to a setting in which the available labels are coarse or indirect. Third, inaccurate supervision occurs when some labels may be incorrect. Semi-supervised learning is a form of incomplete supervision. It is often formulated under the manifold assumption.

A classical example is manifold regularization, which assumes that similar data points should have similar predictions, even when their labels are unavailable (Belkin et al., 2006):

$$\underbrace{\frac{1}{l}\sum_{i=1}^{l} V(x_i, y_i, f)}_{\text{fit to labeled data}} + \underbrace{\gamma_A \|f\|_K^2}_{\text{complexity control in the ambient space}} + \underbrace{\gamma_I \|f\|_I^2}_{\text{smoothness along the data manifold}} . \tag{1}$$

The first term fits the labeled targets, while the second regularizer controls the complexity of f in the ambient space. The third regularizer encodes the manifold assumption that similar inputs should have similar predictions. It therefore imposes a consistency constraint along the data manifold. In semi-supervised regression, this principle can also be applied between target-preserving augmented views. SymRankor reinterprets the ARC used in RankUp as a mechanism for generating potentially noisy pairwise pseudo-labels from unlabeled data (Huang et al., 2024). It transfers the predicted order from a weak-view to a strong-view. Thus, SymRankor combines semi-supervised learning with noisy-label robustness.

Learning to rank. Objectives in this area are conventionally grouped by the unit on which the loss is defined; Table 4 in Appendix A summarizes the three standard families.

Early pairwise work learns preference functions from observed comparisons (Cohen et al., 1998), and RankBoost combines such preferences through boosting (Freund et al., 2003). RankNet instead expresses each pairwise comparison as a Bradley–Terry style probability based on the score difference and optimizes the corresponding cross entropy loss (Bradley & Terry, 1952; Burges et al., 2005). By contrast, listwise approaches represent an entire ranking through a relevance vector or a permutation (Cao et al., 2007; Xia et al., 2008; Taylor et al., 2008). Let K items have real-valued scores $\mathbf{r} = (r_1, \ldots, r_K)$, and let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ order their indices from most to least preferred. The Plackett–Luce probability of a permutation is obtained through successive choices from the remaining items (Luce, 2012; Plackett, 1975) and can be written as

$$p_{\mathrm{PL}}(\boldsymbol{\pi} \mid \mathbf{r}) = \prod_{k=1}^{K-1} \frac{\exp(r_{\pi_k})}{\sum_{j=k}^{K} \exp(r_{\pi_j})}. \tag{2}$$

This gives a likelihood over the full permutation. ListMLE trains a ranker by maximizing it (Xia et al., 2008). For $K = 2$ a single factor remains and the likelihood reduces to $\sigma(r_{\pi_1} - r_{\pi_2})$, with σ the logistic function — the Bradley–Terry probability of Eq. (3). RankUp and SymRankor also use pairwise objectives, but their unlabeled order labels are generated by the model and can therefore be incorrect.

Preference optimization. Pairwise preference data are used to train reward models (Christiano et al., 2017). The learned reward can then guide policy-optimization algorithms such as PPO (Schulman et al., 2017). Offline methods such as DPO instead optimize directly from preference pairs (Rafailov et al., 2023). SymPO casts reward modeling with noisy preferences as binary classification and applies symmetric losses to the pairwise reward margin. Under its preference-noise model, the noisy and clean reward risks share the same minimizers. A classification-calibrated loss then yields a rank-preserving reward, which suffices for policy improvement under the paper's assumptions (Nishimori et al., 2026). SymRankor uses this connection only to motivate a noise-robust pairwise auxiliary objective for semi-supervised regression and makes no policy-optimization claim.

3 Preliminaries and Problem Setup

We adopt RankUp's semi-supervised regression setup. We are given a small labeled set $\mathcal{D}_{\mathrm{lb}} = \{(x_i, y_i)\}_{i=1}^{N_{\mathrm{lb}}}$ with continuous targets $y_i \in \mathbb{R}$ and a large unlabeled set $\mathcal{D}_{\mathrm{ulb}} = \{x_i\}_{i=1}^{N_{\mathrm{ulb}}}$. A backbone with a regression head produces predictions. An auxiliary ranking classifier (ARC) shares the backbone features. As in RankUp, weak and strong augmentations are assumed to preserve the underlying target order. The model is trained to align strong-view predictions with pseudo-labels generated from weak-view predictions (Huang et al., 2024). The constant-flip model introduced below is used only to derive the risk identity. It is not imposed or estimated during training.

Bradley–Terry (BT) model. For two items with real-valued scores r_i, r_j, the BT model (Bradley & Terry, 1952) gives the probability that i is preferred to j as a function of the score difference,

$$p_{\mathrm{BT}}(x_i \succ x_j \mid r_i, r_j) = \sigma(r_i - r_j), \qquad \sigma(u) = \frac{1}{1 + e^{-u}}. \tag{3}$$

In the binary setting, the Plackett–Luce construction specializes to the BT form in Equation (3). The ARC implements this model with a two-dimensional output $a_\theta(x) \in \mathbb{R}^2$ for each sample. For a pair (x_i, x_j), the pairwise logits are obtained by subtracting the corresponding per-item logits. We index the outputs so that class 1 denotes the event $x_i \succ x_j$. This matches the convention enforced by the labeled cross-entropy target. We define the scalar rank score as the logit difference

$$r_i = a_{\theta,1}(x_i) - a_{\theta,0}(x_i). \tag{4}$$

The ARC's pairwise class probabilities are

$$p^v_{ij,c} = \frac{\exp\big(a_{\theta,c}(x_i^v) - a_{\theta,c}(x_j^v)\big)}{\sum_{c' \in \{0,1\}} \exp\big(a_{\theta,c'}(x_i^v) - a_{\theta,c'}(x_j^v)\big)}, \qquad p^v_{ij,1} = \sigma\big(r_i^v - r_j^v\big), \tag{5}$$

where the superscript $v \in \{w, s\}$ marks the weak or the strong view of the inputs. The right-hand identity recovers Eq. (3), so $r_i - r_j$ is the pairwise score difference entering the BT model. On labeled pairs, RankUp trains this head with cross-entropy using the binary comparison target $\mathbf{1}\{y_i > y_j\}$.

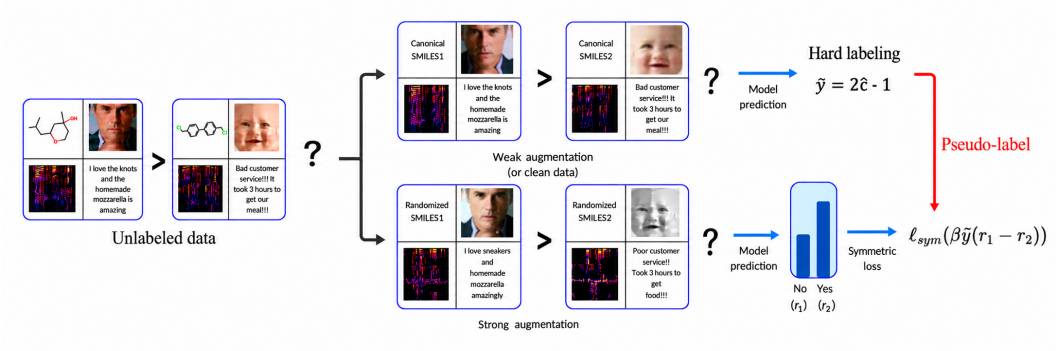


Figure 1: Overview of SymRankor. The detached weak branch produces a pairwise pseudo-label obtained by applying the arg-max and the class-to-sign mapping $\tilde{y} = 2\hat{c} - 1$ of Eq. (7), and the strong branch produces the score margin used for optimization. Labeled pairs retain the RankUp cross-entropy objective, whereas unlabeled pairs use the symmetric loss ℓ_{sym}. Gradients from the unlabeled term flow through the strong branch only.

Symmetric loss. A margin loss ℓ is *symmetric* if there is a constant K with

$$\ell(z) + \ell(-z) = K. \tag{6}$$

Under the constant binary flip model introduced below, this identity shows that the corrupted population risk is a positive affine transformation of the clean risk. SymRankor uses the sigmoid loss $\ell(z) = (1 + e^z)^{-1}$ as its default choice. It is bounded in $[0, 1]$ and satisfies $K = 1$. Ramp and unhinged losses are also symmetric, whereas the logistic (binary cross-entropy) loss is not. Any nonconstant margin loss that is both nonnegative and globally symmetric must be nonconvex. The affine unhinged loss is convex and symmetric but unbounded below (Charoenphakdee et al., 2019). Appendix B gives the exact forms of the standard symmetric losses and describes additional symmetric variants introduced in this work, including Gaussian-based forms.

For an unlabeled pair, the model's predicted order may disagree with the order implied by the unknown targets. Charoenphakdee et al. (2019) analyze symmetric losses under a fixed label-corruption process. SymPO similarly models pairwise preferences with class-conditional flip rates that do not vary across instances (Nishimori et al., 2026). We use the equal-rate case only to derive the population-risk identity. This assumption affects only the theoretical analysis. It does not change the SymRankor training procedure.

For the population-risk analysis, we consider pairwise comparisons with distinct target values and define $y_{ij} = \mathrm{sign}(y_i - y_j) \in \{-1, +1\}$. We fix the pseudo-labeling mechanism. For each pair in this binary-order population, we assume

$$\Pr(\tilde{y}_{ij} = -y_{ij} \mid x_i, x_j, y_{ij}) = \rho, \qquad \rho \in (0, 0.5).$$

This is the input-independent symmetric special case of class-conditional corruption (Natarajan et al., 2013; Nishimori et al., 2026). The condition $\rho < 0.5$ means that the observed order is more likely to be correct than reversed. The rate ρ is used only in this analysis. SymRankor neither injects nor estimates it during training. The risk identity does not require its value.

4 Method

SymRankor retains RankUp's backbone, regression head, and labeled ARC loss. Its unlabeled ARC term uses the symmetric margin loss together with the pair indexing, masking, and normalization specified in Eq. (10). Figure 1 summarizes the pipeline. The shared backbone encodes weakly and strongly augmented views of each input. The weak-view yields a pairwise pseudo-label, whereas the strong-view yields the margin used for backpropagation. The symmetric loss ℓ_{sym} (sigmoid by default) combines them using margin scale β.

Noisy Bradley–Terry interpretation. Eq. (5) shows that the ARC's two-way softmax admits an exact BT-form reparameterization. Here, "noisy Bradley–Terry" refers only to the possibility that

Algorithm 1 SymRankor training step (unlabeled ARC term).

Input: labeled batch, unlabeled batch; weak and strong augmentations A_w, A_s; scale β; optional threshold τ
Encode weak and strong views with the shared backbone; get regression and ARC outputs
$\ell_{\mathrm{reg}} \leftarrow \ell_1$ on labeled predictions; $\ell_{\mathrm{arc_{lb}}} \leftarrow$ cross-entropy on labeled pairs
for each ordered unlabeled pair (i, j), $i \neq j$ **do**
    $\hat{c}_{ij}^{w} \leftarrow \arg\max_{c\in\{0,1\}} p_{ij,c}^{w}$ without gradient; $\tilde{y}_{ij} \leftarrow 2\hat{c}_{ij}^{w} - 1$ ▷ Eqs. (5)–(7)
    $q_{ij}^{w} \leftarrow \max_c p_{ij,c}^{w}$; $m_{ij} \leftarrow \mathbf{1}\{q_{ij}^{w} \geq \tau\}$ ▷ $\tau = 0$ keeps every pair
    $\ell_{ij} \leftarrow \ell_{\mathrm{sym}}\big(\beta\, \tilde{y}_{ij}(r_i^s - r_j^s)\big)$ ▷ Eq. (9)
end for
$\ell_{\mathrm{arc_{ulb}}} \leftarrow$ mask-weighted mean of ℓ_{ij}, taken as 0 if no pair is retained ▷ Eq. (10)
Update on ℓ_{total} ▷ Eq. (8)

the detached hard pseudo-order is corrupted. The method does not introduce an explicit noisy BT likelihood or estimate a label-flip rate. For an unlabeled pair (x_i, x_j), let $\hat{c}_{ij}^{w} \in \{0, 1\}$ denote the corresponding detached arg-max of the ARC predictions on the weakly augmented views. We convert this class index to a signed pseudo-order,

$$\tilde{y}_{ij} = 2\hat{c}_{ij}^{w} - 1, \tag{7}$$

so that $\tilde{y}_{ij} = +1$ denotes the predicted order $x_i \succ x_j$. This matches the class convention of Eq. (5) and the binary-order convention used in the analysis. This mapping retains only the predicted order direction. The symmetric margin loss requires this signed pseudo-label and the strong-view score difference.

Objective. The training loss mirrors RankUp,

$$\ell_{\mathrm{total}} = \ell_{\mathrm{reg}} + w_{\mathrm{arc_{lb}}}\, \ell_{\mathrm{arc_{lb}}} + w_{\mathrm{arc_{ulb}}}\, \ell_{\mathrm{arc_{ulb}}}, \tag{8}$$

where ℓ_{reg} is an ℓ_1 regression loss and $\ell_{\mathrm{arc_{lb}}}$ is cross-entropy on labeled pairs, both identical to RankUp. The only change is the unlabeled ARC term $\ell_{\mathrm{arc_{ulb}}}$, which uses the symmetric loss. Writing r_i^s for the score of Eq. (4) evaluated on the strongly augmented i-th sample, and $\tilde{y}_{ij}$ for the pairwise pseudo-label of Eq. (7), the unlabeled ARC term is defined as

$$\ell_{ij} = \ell_{\mathrm{sym}}\big(\beta\, \tilde{y}_{ij}\, (r_i^s - r_j^s)\big)\,, \tag{9}$$

$$\ell_{\mathrm{arc_{ulb}}} = \frac{\sum_{i\neq j} m_{ij}\, \ell_{ij}}{\max\Big(\sum_{i\neq j} m_{ij},\ 1\Big)}, \tag{10}$$

where β is the margin scale and $m_{ij} \in \{0, 1\}$ masks the ordered pairs that are kept. Let $q_{ij}^{w} = \max_{c\in\{0,1\}} p_{ij,c}^{w}$ denote the weak-view confidence. For $i \neq j$, define $m_{ij} = \mathbf{1}\{q_{ij}^{w} \geq \tau\}$. The choice $\tau = 0$ assigns $m_{ij} = 1$ to every non-self pair. If $\tau > 0$, it retains only pairs that meet the weak-view confidence threshold. The clamped denominator keeps the term well-defined when masking removes all pairs. The training step is given as Algorithm 1. The remaining implementation details are provided in Appendix D.

To formalize why the symmetric term tolerates flipped pairs, consider the binary-order model introduced earlier (Charoenphakdee et al., 2019; Nishimori et al., 2026). Treat each sample pair as one input and let the classifier margin be the scaled score difference used in Eq. (9),

$$g_\beta(x_i, x_j) = \beta\,(r_i - r_j), \qquad \beta > 0. \tag{11}$$

Define the clean and noisy risks of the unlabeled pairwise term as

$$R_{\mathrm{clean}}(g_\beta) = \mathbb{E}_{x,y}[\ell(y g_\beta)], \qquad R_{\mathrm{noisy}}(g_\beta) = \mathbb{E}_{x,y,\tilde{y}}[\ell(\tilde{y} g_\beta)]. \tag{12}$$

Under a constant flip rate $\rho \in (0, 0.5)$, conditioning on whether the pseudo-label is flipped and applying the symmetric condition in Eq. (6) yields

$$R_{\mathrm{noisy}}(g_\beta) = (1 - 2\rho)\, R_{\mathrm{clean}}(g_\beta) + \rho K. \tag{13}$$

Because $1 - 2\rho > 0$ and ρK is independent of g_β, Eq. (13) is a positive affine transformation of R_{clean}. It therefore preserves risk ordering. When the risks are well-defined and their minima are attained, they share the same argmin set. The derivation and its assumptions are given in Appendix C.

Pseudo-ordinal annotation. Coarse domain knowledge can be incorporated because the ARC requires only order information. Using labeled data, we select and orient a descriptor correlated with the target. On the unlabeled pool, its top and bottom fractions of size r define two ordered classes. We evaluate $r \in \{0.1, 0.2\}$ in the chemical case study. This procedure measures no additional continuous targets. Figure 2 and Algorithm 2 in Appendix E give the complete construction.

The pseudo-ordinal construction introduces two additional terms in the objective:

$$\ell_{\text{total}} = \ell_{\text{reg}} + w_{\text{arc}_{\text{lb}}}\ell_{\text{arc}_{\text{lb}}} + w_{\text{arc}_{\text{ulb}}}\ell_{\text{arc}_{\text{ulb}}} + \underbrace{w_{\text{cls}_{\text{diff}}}\ell_{\text{cls}_{\text{diff}}} + w_{\text{cls}_{\text{same}}}\ell_{\text{cls}_{\text{same}}}}_{\text{pseudo-ordinal term}}, \tag{14}$$

Both are evaluated with the symmetric loss. The two pseudo-ordinal terms use complementary signals. For a pair assigned to different pseudo-ordinal classes, $\ell_{\text{cls}_{\text{diff}}}$ uses the class order as cross-class ordinal supervision and evaluates the symmetric loss on the canonical-view margin. For pairs assigned to the same pseudo-ordinal class, $\ell_{\text{cls}_{\text{same}}}$ enforces within-class pairwise consistency.

In the chemical case study, descriptor-derived class $0/1$ labels provide coarse or inexact ordinal supervision for the continuous regression target. Because the descriptor does not determine every target ordering correctly, these labels may also contain instance-level ordering errors. Therefore, SymRankor with pseudo-ordinal annotation is a weakly supervised regression method with incomplete, inexact, and inaccurate supervision.

5 Experimental Setup

Our evaluation has three parts. We first compare SymRankor with classification-derived and regression-specific semi-supervised baselines across three modalities. We then cross the symmetric loss with the confidence threshold to assess their separate empirical effects. A chemical case study finally examines whether coarse pseudo-ordinal annotation improves prediction on a scientific target.

Metrics and benchmarks. We evaluate on the RankUp benchmark, comprising UTKFace for vision, BVCC for audio, and Yelp Review for text (Huang et al., 2024). For each task, 250 training examples are labeled and the remainder of the training pool is treated as unlabeled. Our experiments use the USB-based protocol adopted by RankUp for data splits, backbone architectures, optimizers, learning-rate schedules, training budgets, and model selection (Huang et al., 2024; Wang et al., 2022). We report MAE (lower is better), R^2, and SRCC (higher is better), as mean $\pm$ standard deviation over seeds $0, 1, 2$. Appendix G lists the prediction target, data split, evaluation split, labeled training set size, and backbone for each task. Appendix D describes the dataset-specific augmentation choices. It also reports the GPU hours required for one seed on each benchmark task. Appendix G explains SMILES and the representations used in the chemical case study.

Baselines. The comparison spans three groups with distinct roles. The classification-derived methods (Π-Model (Laine & Aila, 2017), Mean Teacher (Tarvainen & Valpola, 2017), and MixMatch (Berthelot et al., 2019)) were originally developed for semi-supervised classification. Their regression adaptations are evaluated within the same setup. Regression-specific methods provide the competitive reference. CLSS derives ordinal relations through spectral seriation (Dai et al., 2023a), UCVME combines uncertainty consistency with variational ensembling (Dai et al., 2023b), and SSR-RCUS uses manifold mixup (Verma et al., 2019) to preserve ranking relations between nearby unlabeled examples (Li et al., 2026). RankUp is the direct predecessor. Table 1 also reports the original results alongside those enhanced by Regression Distribution Alignment (RDA). RDA refines the distribution of regression pseudo-labels without modifying the ARC loss (Huang et al., 2024). Each baseline keeps its published method-specific configuration within the shared protocol. The labeled-only and fully supervised rows are contextual references rather than competing semi-supervised methods.

Except for the SymRankor-specific quantities, the benchmark-level training settings follow the inherited protocol. We tune only these quantities under the same model-selection convention. The search uses Optuna's implementation of the Tree-structured Parzen Estimator (Bergstra et al., 2011; Akiba et al., 2019). Appendix H reports the selected values and the shared per-task settings. To be

Table 1: Semi-supervised regression on Text (Yelp, sentiment), Audio (BVCC, quality), and Vision (UTKFace, age). The *Supervised* and *Fully supervised* rows are contextual references rather than competing methods. Among the semi-supervised methods, boldface marks the best mean and nearties within half a unit in the last reported digit. The primary SymRankor configuration uses $\tau = 0$ and hyperparameter optimization (HPO). Other SymRankor rows are secondary variants analyzed in Section 6.2.

	Text (Yelp)			Audio (BVCC)			Vision (UTKFace)		
Method	MAE ↓	R^2 ↑	SRCC ↑	MAE ↓	R^2 ↑	SRCC ↑	MAE ↓	R^2 ↑	SRCC ↑
Supervised	0.728±0.026	0.548±0.029	0.752±0.014	0.527±0.005	0.505±0.015	0.743±0.021	9.620±0.296	0.520±0.026	0.699±0.010
Π-Model (2017.04)	0.735±0.024	0.560±0.017	0.761±0.010	0.526±0.005	0.504±0.018	0.742±0.022	9.545±0.327	0.512±0.031	0.705±0.025
Mean Teacher (2017.12)	0.736±0.026	0.550±0.023	0.761±0.010	0.527±0.006	0.505±0.018	0.745±0.017	8.858±0.203	0.591±0.019	0.746±0.010
MixMatch (2019.12)	0.888±0.008	0.380±0.016	0.655±0.011	0.584±0.011	0.385±0.026	0.643±0.031	7.911±0.160	0.692±0.015	0.829±0.006
UCVME (2023.02)	0.775±0.009	0.539±0.008	0.756±0.009	0.491±0.005	0.564±0.009	0.768±0.005	8.732±0.243	0.611±0.027	0.761±0.019
CLSS (2023.12)	0.717±0.021	0.524±0.026	0.732±0.015	0.461±0.011	0.606±0.015	0.777±0.008	9.086±0.089	0.580±0.006	0.734±0.006
RankUp (2024.12)	0.653±0.018	0.654±0.012	0.834±0.001	0.458±0.012	0.612±0.018	0.785±0.011	7.026±0.141	0.751±0.006	0.833±0.001
RankUp + RDA (2024.12)	0.597±0.020	0.636±0.022	0.815±0.011	0.453±0.008	0.617±0.013	0.788±0.007	6.370±0.099	0.796±0.006	0.862±0.002
SSR-RCUS (2026.01)	0.616±0.030	0.678±0.027	0.831±0.012	0.464±0.011	0.598±0.022	0.779±0.011	**6.340±0.062**	**0.803±0.007**	**0.863±0.005**
SymRankor (τ=0)	0.580±0.013	0.672±0.009	**0.848±0.006**	0.454±0.010	0.614±0.010	0.788±0.003	7.060±0.141	0.749±0.013	0.831±0.007
SymRankor (τ=0.95)	0.576±0.009	0.676±0.004	0.833±0.003	0.450±0.011	0.622±0.013	0.790±0.012	7.005±0.056	0.754±0.006	0.836±0.001
SymRankor (τ=0, HPO)	**0.564±0.015**	0.708±0.008	0.846±0.002	**0.428±0.005**	**0.654±0.007**	**0.810±0.005**	6.411±0.032	0.797±0.006	0.851±0.002
SymRankor (τ=0.95, HPO)	**0.563±0.015**	**0.710±0.009**	0.847±0.003	0.430±0.004	0.652±0.006	**0.810±0.005**	6.883±0.128	0.760±0.008	0.834±0.013
Fully supervised	0.421±0.004	0.793±0.003	0.893±0.001	0.336±0.003	0.785±0.003	0.886±0.003	4.797±0.017	0.877±0.001	0.912±0.001

fair, Table 1 reports tuned and default SymRankor configurations at $\tau = 0$ and $\tau = 0.95$. The comparison shows the effects of method-specific tuning and confidence filtering. We further evaluate the symmetric loss and standard cross-entropy objective in a 2×2 ablation. Each objective is tested at $\tau \in \{0, 0.95\}$ with all other hyperparameters fixed. Section 6.2 reports the results.

AqSolDB is a curated reference set of measured aqueous solubilities used for the chemical case study (Sorkun et al., 2019). The dataset is accessed through the Therapeutics Data Commons (Huang et al., 2021). A single molecular-scaffold split divides the dataset into training, validation and test sets in an 8:1:1 ratio. Each seed uses 250 molecules as labeled examples and treats the rest of the training split as unlabeled. Training uses the text-task configuration. MolFormer serves as the molecular encoder (Ross et al., 2022). An RDKit-computed molecular descriptor defines the two ordinal classes through its lower and upper tails. This case study uses LogP with bound ratios $r \in \{0.1, 0.2\}$. The descriptor values come from molecular structure alone, whereas descriptor selection and orientation use the labeled training set. Appendix E details the pseudo-ordinal construction and presents Algorithm 2. Appendix F reports the descriptor–target correlation. Dataset details appear in Appendix G. Appendix H lists the hyperparameters.

6 Results

6.1 Main comparison

To assess the practical value of SymRankor, Table 1 compares SymRankor with semi-supervised baselines on the three benchmark tasks. Each baseline retains its published method-specific configuration within the shared protocol. We report two SymRankor configurations. One uses the default settings and the other applies hyperparameter optimization (HPO).

The default configuration adopts the margin-based sigmoid loss. It otherwise retains the experimental settings of RankUp. This enables a fair comparison with our most direct predecessor. Even under this condition the default configuration remains competitive. At $\tau = 0$, the default configuration reduces text MAE by 11.2% relative to RankUp. It reaches 0.454 on audio and 7.060 on vision. Both results remain close to RankUp. The thresholded default lowers MAE slightly on all three tasks.

The primary configuration results from HPO at $\tau = 0$. It tunes only quantities introduced by SymRankor under the same validation protocol. It reaches an MAE of 0.564 on text. It reaches 0.428 on audio and 6.411 on vision. Its MAE improves on RankUp for every task. It also improves on RankUp + RDA on text and audio. The tuned configuration with $\tau = 0.95$ gives comparable results on text and audio. Its vision MAE is weaker at 6.883.

Table 2: Ablation over the symmetric loss and the confidence threshold τ on the pairwise pseudo-label, at the tuned setting. ✓ uses the symmetric loss, ✗ replaces it with the cross-entropy objective of RankUp. All four cells share the same non-loss hyperparameters. Within each modality, bold marks the best mean per metric and any mean within half of the last reported digit of it.

		Text (Yelp)			Audio (BVCC)			Vision (UTKFace)		
Sym. loss	τ	MAE ↓	R^2 ↑	SRCC ↑	MAE ↓	R^2 ↑	SRCC ↑	MAE ↓	R^2 ↑	SRCC ↑
✓	0	**0.564**$\pm$**0.015**	0.708$\pm$0.008	0.846$\pm$0.002	**0.428**$\pm$**0.005**	**0.654**$\pm$**0.007**	**0.810**$\pm$**0.005**	**6.411**$\pm$**0.032**	**0.797**$\pm$**0.006**	**0.851**$\pm$**0.002**
✗	0	0.603$\pm$0.021	0.700$\pm$0.016	**0.848**$\pm$**0.006**	0.462$\pm$0.009	0.606$\pm$0.010	0.778$\pm$0.007	6.845$\pm$0.237	0.776$\pm$0.011	0.847$\pm$0.007
✓	0.95	**0.563**$\pm$**0.015**	**0.710**$\pm$**0.009**	0.847$\pm$0.003	0.430$\pm$0.004	0.652$\pm$0.006	**0.810**$\pm$**0.005**	6.883$\pm$0.128	0.760$\pm$0.008	0.834$\pm$0.013
✗	0.95	0.600$\pm$0.022	0.699$\pm$0.019	0.847$\pm$0.005	0.453$\pm$0.011	0.616$\pm$0.015	0.786$\pm$0.010	6.851$\pm$0.351	0.770$\pm$0.018	0.842$\pm$0.013

SSR-RCUS remains ahead of SymRankor on vision. UTKFace spans the widest target range among the three benchmarks. Its use of manifold mixup follows the manifold assumption of Belkin et al. (2006). Similar inputs should have similar predictions. For unlabeled pairs SSR-RCUS creates nearby latent samples. It transfers the inferred order of the original pair. This may help age prediction because order preservation can be more reliable when the target gap is large.

6.2 Where the performance gain comes from

Table 2 isolates the effects of the symmetric loss and confidence filtering. All hyperparameters are fixed except for the loss and τ. Each setting starts from the same non-self pair set. At $\tau = 0$ the symmetric loss lowers MAE on every task. The symmetric loss reduces MAE by 6.5% on text. The reductions are 7.4% on audio and 6.3% on vision. At $\tau = 0.95$ the symmetric loss still lowers MAE and raises R^2 on text and audio. Text SRCC changes little across the settings. Vision behaves differently. The symmetric loss performs best without filtering and reaches an MAE of 6.411 at $\tau = 0$. This suggests that the symmetric loss can use noisy pseudo-orders without hard confidence filtering.

Strong performance does not require confidence filtering. With the symmetric loss fixed the MAE changes by 0.001 on text and 0.002 on audio when τ increases from 0 to 0.95. The setting with $\tau = 0$ uses all non-self pairs throughout training. A high threshold can delay the use of unlabeled data. Appendix I.1 examines this issue through an ablation of the Gaussian loss on vision. Figure 4 shows one run whose gate admits no unlabeled pair until 48.4% of training has elapsed. Its consistency term stays inactive for almost half of the schedule. Tables 11 and 12 show the resulting performance drop and instability. The Gaussian configuration with $\tau = 0.95$ has a higher MAE and a wider seed spread than the configuration with $\tau = 0$. The cross-entropy arm shows that this risk is not confined to the Gaussian loss. Its slowest-opening seed began to use unlabeled pairs substantially only at about 10% of training. It reached an MAE of 7.1522, above 6.5785 and 7.0129 for the two earlier-opening seeds. Early confidence filtering can therefore prevent the model from using unlabeled data when that signal is most needed.

These results suggest that hard confidence filtering may delay gains from unlabeled data when it prioritizes pseudo-label quality. FixMatch's strict confidence threshold can leave useful unlabeled examples unused. SoftMatch instead uses confidence-dependent weights to involve more unlabeled data and improve performance (Chen et al., 2023). In our ablation, the hard gate ($\tau = 0.95$) can leave the unlabeled term inactive early even though roughly 90% of pairs pass it by the end. This suggests that early suppression of unlabeled information can itself become a performance bottleneck. SymRankor avoids this scheduling cost at $\tau = 0$ by relying on the symmetric loss to limit the effect of incorrect pseudo-orders. It would therefore be beneficial to develop diverse symmetric losses and low-cost annotation methods.

Table 13 compares additional symmetric losses under the default vision configuration at $\tau = 0$. Gaussian gives the best mean on all three metrics among these losses. The method also has a broader implication. Every retained unlabeled pair provides a pseudo-label from its weak-view. SymRankor can therefore use cheap ordinal labels at a much larger scale than the observed continuous targets. Section 6.3 examines a case for cheap labeling. It asks whether a user's domain expertise can provide ordinal guidance that improves prediction without collecting additional costly continuous values.

Table 3: Chemical case study on AqSolDB. The pseudo-ordinal term assigns the extremes of the LogP descriptor to two ordinal classes at bound ratio r; the plain row adds no such term. Best per metric in bold.

Method	r	MAE ↓	R^2 ↑	SRCC ↑
SymRankor	–	$1.033_{\pm 0.061}$	$0.605_{\pm 0.038}$	$0.769_{\pm 0.027}$
+ pseudo-ordinal	0.1	$1.025_{\pm 0.067}$	$0.610_{\pm 0.045}$	$0.775_{\pm 0.033}$
+ pseudo-ordinal	0.2	$\mathbf{1.018}_{\pm \mathbf{0.041}}$	$\mathbf{0.614}_{\pm \mathbf{0.031}}$	$\mathbf{0.779}_{\pm \mathbf{0.017}}$

6.3 Chemical case study

The AqSolDB case study evaluates whether SymRankor can use LogP as coarse and imperfect ordinal information about aqueous solubility. Measuring aqueous solubility normally requires laboratory work. RDKit computes LogP from molecular structure alone (RDKit, 2025). Appendix E gives the full pseudo-ordinal construction and Figure 2 shows how the upper and lower tails form the ordinal classes. Appendix F reports Spearman correlations between LogP and solubility from -0.79 to -0.71 across the three labeled splits. These correlations support its use for relative solubility information but do not imply that every descriptor-induced comparison is correct.

Table 3 shows that the pseudo-ordinal term improves the mean of every metric. At $r = 0.2$ MAE falls from 1.033 to 1.018. R^2 rises from 0.605 to 0.614. SRCC rises from 0.769 to 0.779. The gain also grows from $r = 0.1$ to $r = 0.2$. The result is consistent with basic chemical knowledge. LogP measures lipophilicity, which is generally negatively associated with aqueous solubility. These results show that LogP computed from molecular structure can improve continuous solubility prediction despite imperfect ordinal relations.

7 Limitations

SymRankor's Bradley–Terry formulation models pairwise order rather than a distribution over complete rankings. How a listwise target interacts with inconsistent model-generated pseudo-orders remains untested here. Future work could complement the pairwise ARC with an RDA-style distribution-level loss that aligns aggregate unlabeled predictions with the labeled target distribution without imposing hard itemwise pseudo-labels.

A second issue is that pseudo-ordinal annotation is demonstrated only in the chemical case study. Although we explored rule-based tools for the other three modalities, we did not identify a comparably systematic domain principle for deriving ordinal supervision. One possible direction is to explore pairwise judgments from human users or domain experts who do not observe the continuous target values.

8 Conclusion

We reinterpret the unlabeled auxiliary ranking objective in semi-supervised regression as learning from potentially corrupted pairwise pseudo-labels. SymRankor replaces cross-entropy on unlabeled pairs with a symmetric margin loss. Under a constant binary flip model, this loss preserves the clean-risk minimizers without requiring the flip rate. Empirically, SymRankor lowers MAE on all three tasks relative to cross-entropy. It remains competitive with the thresholded setting while using every non-self pair. SymRankor thus avoids trading early unlabeled-pair use for hard confidence filtering. A molecular-solubility case study further shows how the same ordinal objective can incorporate coarse domain-derived order information.

More broadly, since value-function approximation can be cast as regression (Sugiyama, 2015, Ch. 1, p. 13), SymRankor may provide a starting point for future work in reinforcement learning.

Reproducibility Statement

Appendices C–H cover the assumptions and proofs of the theoretical claims, the implementation details, the data construction, and the hyperparameters. The supplementary material provides the anonymized SymRankor source code, its configuration files, and the prepared molecular data.

AI Use Statement

We used generative AI tools for literature retrieval and discovery, research ideation, and drafting and polishing portions of this manuscript. We also used them to assist with exploring mathematical claims and proof ideas. All AI-assisted outputs were reviewed and revised by the authors, and the final mathematical claims and proofs were checked by the authors. We take responsibility for the final content of this paper, including all text, claims, and artifacts produced with the aid of generative AI.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2623–2631, 2019. doi: 10.1145/3292500.3330701.

Josep Arús-Pous, Simon Viet Johansson, Oleksii Prykhodko, Esben Jannik Bjerrum, Christian Tyrchan, Jean-Louis Reymond, Hongming Chen, and Ola Engkvist. Randomized SMILES strings improve the quality of molecular generative models. *Journal of Cheminformatics*, 11:71, 2019. doi: 10.1186/s13321-019-0393-0.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019.

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

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

Nontawat Charoenphakdee, Jongyeong Lee, and Masashi Sugiyama. On symmetric losses for learning from corrupted labels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 2019.

Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. In *International Conference on Learning Representations*, 2023.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.

William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. In *Advances in Neural Information Processing Systems*, 1998.

Weihang Dai, Yao Du, Hanru Bai, Kwang-Ting Cheng, and Xiaomeng Li. Semi-supervised contrastive learning for deep regression with ordinal rankings from spectral seriation. In *Advances in Neural Information Processing Systems*, 2023a.

Weihang Dai, Xiaomeng Li, and Kwang-Ting Cheng. Semi-supervised deep regression with uncertainty consistency and variational model ensembling via bayesian neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023b.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W. Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Pin-Yen Huang, Szu-Wei Fu, and Yu Tsao. Rankup: Boosting semi-supervised regression with an auxiliary ranking classifier. In *Advances in Neural Information Processing Systems*, 2024.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017.

Ximing Li, Jiaxuan Jiang, Changchun Li, You Lu, and Renchu Guan. Semi-supervised regression by preserving ranking relationships between close unlabeled samples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

R. Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. Courier Corporation, 2012. ISBN 9780486153391. URL `https://books.google.com/books?id=ERQsKkPiKkkC`.

Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, 2013.

Soichiro Nishimori, Yu-Jie Zhang, Thanawat Lodkaew, and Masashi Sugiyama. On symmetric losses for policy optimization with noisy preferences. *Transactions on Machine Learning Research*, 2026. `https://openreview.net/forum?id=cBWGLmSeao`.

R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):193–202, 1975. doi: 10.2307/2346567.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 28492–28518, 2023.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

RDKit. RDKit: Open-source cheminformatics. `https://www.rdkit.org`, 2025. Version 2025.03.5. Accessed: 2026-09-07.

Jerret Ross, Brian Belgodere, Vijil Chenthamarakshan, Inkit Padhi, Youssef Mroueh, and Payel Das. Large-scale chemical language representations capture molecular structure and properties. *arXiv preprint arXiv:2106.09553*, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 2020.

Murat Cihan Sorkun, Abhishek Khetan, and Süleyman Er. AqSolDB, a curated reference set of aqueous solubility and 2D descriptors for a diverse set of compounds. *Scientific Data*, 6:143, 2019. doi: 10.1038/s41597-019-0151-1.

Masashi Sugiyama. *Statistical Reinforcement Learning: Modern Machine Learning Approaches*. CRC Press, 2015.

Masashi Sugiyama, Han Bao, Takashi Ishida, Nan Lu, Tomoya Sakai, and Gang Niu. *Machine Learning from Weak Supervision: An Empirical Risk Minimization Approach*. MIT Press, Cambridge, Massachusetts, USA, 2022.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, 2017.

Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. Softrank: Optimizing non-smooth rank metrics. In *Proceedings of the International Conference on Web Search and Data Mining*, 2008.

Taira Tsuchiya, Nontawat Charoenphakdee, Issei Sato, and Masashi Sugiyama. Semi-supervised ordinal regression based on empirical risk minimization. *arXiv preprint arXiv:1901.11351*, 2021.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 2019.

Yidong Wang, Hao Chen, Yue Fan, Wang Sun, Ran Tao, Wenxin Hou, Renjie Wang, Linyi Yang, Zhi Zhou, Lan-Zhe Guo, Heli Qi, Zhen Wu, Yu-Feng Li, Satoshi Nakamura, Wei Ye, Marios Savvides, Bhiksha Raj, Takahiro Shinozaki, Bernt Schiele, Jindong Wang, Xing Xie, and Yue Zhang. USB: A unified semi-supervised learning benchmark for classification. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Jiaheng Wei, Hangyu Liu, Tongliang Liu, Gang Niu, Masashi Sugiyama, and Yang Liu. To smooth or not? when label smoothing meets noisy labels. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 2022.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1): 44–53, 2018. doi: 10.1093/nsr/nwx106.

A Learning-to-rank formulations

Table 4 groups learning-to-rank objectives by the unit on which the loss is defined. This expands the summary in the Learning to Rank discussion of Section 2. SymRankor's auxiliary ranking classifier is pairwise because each loss is computed from two items.

Table 4: Learning-to-rank formulations grouped by the unit on which the loss is defined.

Formulation	Unit over which the loss is defined	Typical examples
Pointwise	single item x_i with target or relevance label y_i	least-squares regression, relevance classification, ordinal regression
Pairwise	an item pair (x_i, x_j) and the corresponding order label y_{ij}	RankNet, RankBoost
Listwise	query-level item list with a relevance vector or permutation	ListNet, ListMLE, SoftRank

B SYMMETRIC LOSS FUNCTIONS

All losses below act on the scaled margin $z = \beta\,\tilde{y}\,(r_i - r_j)$ with $\tilde{y} \in \{-1, +1\}$. The sigmoid, ramp, and unhinged losses satisfy the global symmetric condition $\ell(z) + \ell(-z) = K$ of Eq. (6). Barrier-hinge is included separately as a convex, nonnegative alternative that satisfies the condition only on a bounded margin interval and is not used in the reported experiments. The default in SymRankor is the sigmoid loss.

- **Sigmoid** (default): $\ell(z) = 1/(1 + e^z)$, with $K = 1$, bounded in $[0, 1]$, nonconvex.
- **Ramp**: $\ell(z) = \mathrm{clamp}\big(\frac{1}{2} - \frac{1}{2}z,\, 0,\, 1\big)$, with $K = 1$, bounded in $[0, 1]$, nonconvex, and with zero gradient for $|z| > 1$.
- **Unhinged**: $\ell(z) = 1 - z$, with $K = 2$; affine and therefore convex, but unbounded below.
- **Barrier-hinge**: $\ell(z) = \max\big(-10(1 + z) + 1,\ \max(10(z - 1),\, 1 - z)\big)$, convex and nonnegative, and symmetric only on a bounded subset of its domain rather than everywhere. This is the illustrative parameter instance used here, not the unique definition; the cited source states a general family.

These properties are not accidents of the particular functions. Any nonconstant, nonnegative, globally symmetric margin loss must be nonconvex, which is why the bounded choices above are nonconvex and why the one convex symmetric loss listed, the affine unhinged loss, pays for convexity by being unbounded below. The barrier-hinge loss was proposed to recover convexity and nonnegativity together by giving up symmetry outside a bounded region (Charoenphakdee et al., 2019). The logistic (binary cross-entropy) loss is convex and nonnegative but does not satisfy the constant-sum symmetric condition.

Additional symmetric losses. The four entries above are the losses supplied by the cited sources. SymRankor also implements five further symmetric losses, listed here as design variants. They all satisfy $\ell(z) + \ell(-z) = 1$ and are bounded in $[0, 1]$. The reported experiments use the sigmoid loss, so these five are documented rather than evaluated. To specify the transition scale, write the unscaled signed margin as $m = \tilde{y}\,(r_i - r_j)$. The scaled margin used in this appendix is $z = \beta m$. Each loss below acts on the normalized argument z.

- **Cauchy**: $\ell(z) = \frac{1}{2} - \frac{1}{\pi}\arctan(z)$. Symmetric because $\arctan$ is odd. The tail is polynomial, $\ell(z) \sim 1/(\pi z)$ as $z \to \infty$.
- **Gaussian**: $\ell(z) = 1 - \Phi(z) = \frac{1}{2}\,\mathrm{erfc}\big(z/\sqrt{2}\big)$, with Φ the standard normal distribution function. Symmetric because $\Phi(-z) = 1 - \Phi(z)$.
- **Soft-sign**: $\ell(z) = \frac{1}{2}\big(1 - z/\sqrt{1 + z^2}\big)$. Symmetric because $z/\sqrt{1 + z^2}$ is odd and bounded.
- **Laplace**: $\ell(z) = \frac{1}{2}e^{-z}$ for $z \geq 0$ and $\ell(z) = 1 - \frac{1}{2}e^{z}$ for $z < 0$. The two branches give $\ell(0) = \frac{1}{2}$ exactly and sum to one under $z \mapsto -z$.
- **Smooth-ramp**: $\ell(z) = \frac{1}{2}\big[1 - h(z)\big]$ with $h(z) = -1$ for $z \leq -1$, $h(z) = (3z - z^3)/2$ for $|z| < 1$, and $h(z) = 1$ for $z \geq 1$. Symmetric because h is odd. The loss reaches 1 and 0 exactly at $z = \mp 1$.

The Gaussian loss uses Φ, the standard normal distribution function. Here, erfc denotes the complementary error function. It is defined as the complement of the error function.

$$\mathrm{erfc}(x) = 1 - \mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2}\, dt.$$

C Derivation of the noisy-risk identity

We derive Eq. (13) from the two risks in Eq. (12). Assume that ℓ satisfies the symmetry condition in Eq. (6) with constant K. Define $z = y\, g_\beta$, where $y \in \{-1, +1\}$ and g_β is the scaled score difference defined in Eq. (11). Let $\tilde{y}$ denote the observed label obtained by flipping y with probability $\rho \in (0, 0.5)$,

$$\tilde{y} = \begin{cases} y, & \text{with probability } 1-\rho, \\ -y, & \text{with probability } \rho. \end{cases}$$

Taking the expectation over the label-flip process gives

$$\begin{aligned} R_{\mathrm{noisy}}(g_\beta) &= \mathbb{E}_{x,y,\tilde{y}}[\ell(\tilde{y} g_\beta)] = (1-\rho)\, \mathbb{E}_{x,y}[\ell(y g_\beta)] + \rho\, \mathbb{E}_{x,y}[\ell(-y g_\beta)] && (15) \\ &= (1-\rho)\, \mathbb{E}_{x,y}[\ell(y g_\beta)] + \rho\, \mathbb{E}_{x,y}\big[K - \ell(y g_\beta)\big] && \text{(by Eq. (6))} \quad (16) \\ &= (1-2\rho)\, \mathbb{E}_{x,y}[\ell(y g_\beta)] + \rho K = (1-2\rho)\, R_{\mathrm{clean}}(g_\beta) + \rho K. && (17) \end{aligned}$$

Since $0 < \rho < 0.5$, we have $1 - 2\rho > 0$. The term ρK is constant with respect to g_β. Thus, Eq. (13) defines a strictly increasing affine map from the clean risk to the noisy risk. It therefore preserves the ordering of hypotheses by risk. If both minima are attained over the hypothesis class and both expectations are finite, the two risks have the same argmin set:

$$\arg\min_{g_\beta} R_{\mathrm{noisy}}(g_\beta) = \arg\min_{g_\beta} R_{\mathrm{clean}}(g_\beta).$$

In SymRankor, each input to the pairwise objective is a sample pair (x_i, x_j) drawn from the binary-order population. Its margin is $g_\beta(x_i, x_j) = \beta(r_i - r_j)$, where r_i and r_j are the ranking scores computed from Eq. (4). We treat $\beta > 0$ as a fixed hyperparameter that defines the scaled margin used in the symmetric loss. This identity applies only to the unlabeled ARC term. In the full objective of Eq. (8), the shared parameters and the factor $1 - 2\rho$ can change its relative weighting.

We use the constant-flip model as a snapshot approximation. At each update the weak-view pseudo-orders are generated and detached before the strong-view loss is evaluated. They remain fixed during that gradient step. The flip rate is therefore interpreted as a quantity associated with the current update rather than as a fixed property of the dataset. Accordingly, Eq. (13) characterizes the unlabeled ARC term under this local approximation.

D Implementation details

Three details of Algorithm 1 are worth stating precisely. The mask m_{ij} of Eq. (10) is indexed by *ordered* pairs: both (i, j) and (j, i) appear, and the $i = j$ diagonal is removed, so a batch of B unlabeled samples contributes $B(B-1)$ terms. The pairwise pseudo-label of Eq. (7) comes from the weak-view under a stopped gradient, so no gradient from the unlabeled term flows through the weak-view pseudo-label. The denominator of Eq. (10) is clamped at one, which makes the term exactly zero rather than undefined on the early steps where a high threshold retains no pair at all. The labeled ARC term keeps the cross-entropy objective of RankUp unchanged, and the regression head is trained with an ℓ_1 loss.

The labeled term treats ties differently from the unlabeled term, which matters because the benchmark targets contain many of them. It retains all B^2 ordered pairs, including the $i = j$ diagonal, and builds its target from $\mathrm{sign}(y_i - y_j)$. A tied or self pair therefore receives the uniform target $(\frac{1}{2}, \frac{1}{2})$: it contributes a constant to the cross-entropy and no order information, rather than being dropped. The binary-order condition in Section 3 concerns only the population-risk analysis of the unlabeled term. It does not restrict pair construction during training.

Following RankUp (Huang et al., 2024), we generate pseudo-labels from a mildly transformed input and impose consistency on a more strongly transformed input. Both transformations are assumed

to preserve the underlying target order. On UTKFace, random cropping and horizontal flipping form the weak-view. RandAugment forms the strong-view. For BVCC, the weak-view uses random subsampling. The strong-view combines it with random masking, trimming, and padding. The weak Yelp Review input is unchanged. Back-translation generates its strong counterpart.

Hardware and computational cost. All experiments were conducted on a system equipped with eight NVIDIA A100-SXM4-40GB GPUs with 40 GB of memory each. Table 5 reports the wall-clock time for one seed on each benchmark task as the median and observed range across runs. The reported times are measured on a single-GPU basis and represent the time required for one training job using one GPU. The wider range for Audio reflects run-to-run variation. Based on these median single-GPU times, one seed across the three benchmark tasks required approximately 32 hours and 29 minutes.

Table 5: Hardware and wall-clock cost for one seed. Each training job used one NVIDIA A100-SXM4-40GB GPU.

Task	Median time	Range
Vision (UTKFace)	9h 53m	9h 10m–10h 23m
Audio (BVCC)	12h 15m	6h 58m–13h 52m
Text (Yelp)	10h 21m	9h 53m–10h 56m

E PSEUDO-ORDINAL CONSTRUCTION

Figure 2 sketches the construction and Algorithm 2 states the supervision term for pairs from different pseudo-ordinal classes and the consistency regularization term for pairs from the same class.

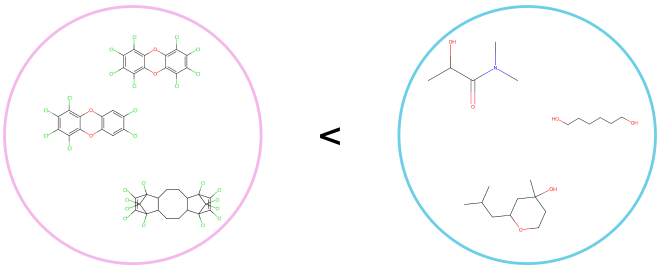


Figure 2: Pseudo-ordinal classes are formed from the two tails of a descriptor computed on the unlabeled pool. A gap between the descriptor ranges increases the likelihood that a cross-class pair is ordered consistently with the target.

RDKit is an open-source cheminformatics toolkit that computes molecular descriptors from molecular structures (RDKit, 2025). We select one descriptor from a bank of RDKit descriptors using its absolute Pearson correlation with the labeled targets. We write this statistic as $\mathrm{corr_P}$ below. This procedure selects LogP for AqSolDB. The same statistic determines the class orientation. Descriptor selection and orientation are the only steps that use labeled targets. These molecules are removed from the unlabeled pool. The ordinal set and the unlabeled ARC term therefore remain disjoint.

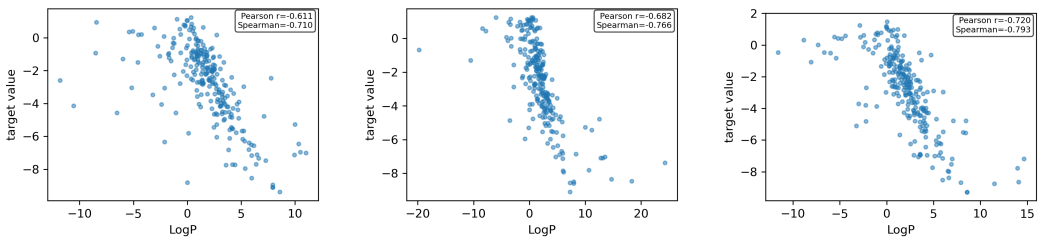
Accordingly, the cross-class term provides cross-class ordinal supervision. The same-class term enforces within-class pairwise consistency. Both terms also use the symmetric loss to handle noise. The same-class term does not assign any descriptor-derived order to same-class pairs.

Algorithm 2 Pseudo-ordinal construction and supervision.

Input: labeled set $\mathcal{D}_{\text{lb}}$, unlabeled pool $\mathcal{D}_{\text{ulb}}$, descriptor bank $\mathcal{F}$, bound ratio r
Construction, once per seed:
Evaluate every $f \in \mathcal{F}$ on $\mathcal{D}_{\text{lb}}$ and compute its Pearson correlation corr_{P} with the labeled targets
$f^\star \leftarrow \arg\max_{f \in \mathcal{F}} |\text{corr}_{\text{P}}(f, y)|$ ▷ LogP on AqSolDB
$v_i \leftarrow f^\star(x_i)$ for $x_i \in \mathcal{D}_{\text{ulb}}$; drop samples whose descriptor is undefined
$n \leftarrow \lfloor r \cdot |\{i : v_i \text{ defined}\}| \rfloor$
$\mathcal{B} \leftarrow$ the n samples with smallest v_i; $\mathcal{T} \leftarrow$ the n samples with largest v_i
if $\text{corr}_{\text{P}}(f^\star, y) < 0$ **then**
    $(\mathcal{C}_0, \mathcal{C}_1) \leftarrow (\mathcal{T}, \mathcal{B})$ ▷ orient so class 0 precedes class 1 in the target
else
    $(\mathcal{C}_0, \mathcal{C}_1) \leftarrow (\mathcal{B}, \mathcal{T})$
end if
$\mathcal{D}_{\text{ulb}} \leftarrow \mathcal{D}_{\text{ulb}} \setminus (\mathcal{C}_0 \cup \mathcal{C}_1)$
Supervision, at each training step:
Draw a balanced batch, half from $\mathcal{C}_0$ and half from $\mathcal{C}_1$; form all ordered pairs and drop self-pairs
$\ell_{\text{cls}_{\text{diff}}} \leftarrow$ symmetric loss on cross-class pairs, using the class-assignment sign and the canonical-view margin ▷ the label is external; no weak-to-strong step is used
for $c \in \{0, 1\}$ **do**
    Restrict to the batch members of class c; take the weak-view arg-max as pseudo-label without gradient and evaluate the symmetric loss on the strong-view margin
end for
$\ell_{\text{cls}_{\text{same}}} \leftarrow$ mean of the two within-class terms
Add both terms to ℓ_{total} ▷ Eq. (14)

F DESCRIPTOR–TARGET CORRELATION

Figure 3 plots the RDKit LogP descriptor against the solubility target for the 250 labeled molecules of each seed. LogP is negatively correlated with solubility, since more lipophilic molecules dissolve less readily. The observed negative association motivates using the extremes of LogP as a coarse proxy for relative solubility, but it does not imply that every descriptor-induced comparison is correct. These two statistics serve different purposes. Descriptor selection and orientation use the Pearson coefficient corr_{P} from Algorithm 2. The values reported below are Spearman rank correlations. This measure is appropriate for supervision that provides only order information.



(a) seed 0 (b) seed 1 (c) seed 2

Figure 3: LogP (RDKit) versus the solubility target for the labeled AqSolDB molecules in each seed. The Spearman rank correlations are consistently negative across seeds with values of -0.71, -0.77, and -0.79.

G DATASETS

Table 6 summarizes the three benchmark tasks. They follow the protocol of RankUp (Huang et al., 2024) which was also adopted by SSR-RCUS (Li et al., 2026). For each task 250 training examples are labeled. The code treats the remainder of each training pool as unlabeled.

UTKFace evaluates face age estimation. Its target is the person's age in years. The metrics therefore measure how closely the predicted age matches the annotated age.

BVCC is the VoiceMOS 2022 dataset for speech quality assessment. Its target is a mean opinion score averaged across ratings from several listeners. The task is to recover this aggregate human judgment.

Yelp Review is used for opinion mining. Its target is the star rating assigned to a written review. The task is to infer the strength of the expressed opinion from text.

All data partitions and evaluation splits follow the public RankUp protocol.

Table 6: Benchmark tasks. Sizes and ranges are those of the RankUp protocol (Huang et al., 2024) and the run configurations.

Task	Target	Range	Train / Eval / Test
UTKFace (vision)	Age in years	$[1, 116]$	18,964 / – / 4,741
BVCC (audio)	Speech quality (MOS)	$[1, 5]$	4,974 / 1,066 / 1,066
Yelp Review (text)	Review rating	$[0, 4]$	250,000 / 25,000 / 10,000

The chemical case study uses **AqSolDB**, a curated reference set of measured aqueous solubilities (Sorkun et al., 2019) obtained through the Therapeutics Data Commons (Huang et al., 2021). Each molecule is represented by a SMILES string. SMILES stands for Simplified Molecular Input Line Entry System. It is a text-based notation that can encode a molecular graph. Canonical SMILES is a deterministic representation produced by a specified canonicalization procedure. We use canonical SMILES as the reference view. The strong-view is generated as a randomized SMILES by changing the atom traversal order. This preserves the underlying molecular graph and chemical validity while providing a different valid input string (Arús-Pous et al., 2019). The target is the logarithm of aqueous solubility. Because the property typically requires experimental measurement, this setting makes a cheaper ordinal signal worth examining. AqSolDB contains $9{,}982$ unique compounds. In our preprocessing, canonicalizing the SMILES strings with RDKit removes two entries that RDKit cannot parse. The remaining $9{,}980$ molecules are split by scaffold into $7{,}984$ training molecules, 998 validation molecules, and 998 test molecules as reported in Table 7. We draw the scaffold split once and keep it fixed. For each seed, 250 molecules from the training partition form the labeled subset.

Table 7: Chemical case study dataset.

Dataset	Target	Ordinal descriptor	Split	Train / Valid / Test
AqSolDB	Aqueous solubility	LogP (RDKit)	Scaffold, 8:1:1	7,984 / 998 / 998

H HYPERPARAMETERS

Table 8 lists the SymRankor-specific parameters of each task. Here `arc_lb` and `arc_ulb` are the weights $w_{\mathrm{arc_{lb}}}$ and $w_{\mathrm{arc_{ulb}}}$ of Eq. (8), `uratio` is the ratio of unlabeled to labeled samples in a batch, and `beta` is the margin scale β of Eq. (9). Where a setting is not listed, $\beta = 1$.

Table 8: SymRankor-specific hyperparameters.

Task	`arc_lb`	`arc_ulb`	`uratio`	`beta`
Vision (UTKFace)	0.7	3.9	11	3.5
Audio (BVCC)	0.4	0.05	4	0.2
Text (Yelp)	0.05	1.0	8	1.2
Chemistry (AqSolDB)	0.2	0.2	7	1.0

The shared training protocol follows the USB standard setting (Wang et al., 2022) that RankUp adopts (Huang et al., 2024), and each baseline keeps its own published method-specific con-

figuration within it. Table 9 lists the per-task settings. The Vision, Audio, and Text columns in Table 9 follow the configurations reported by RankUp. The Chemistry column belongs to our case study and uses the Text configuration. A dash marks a setting the corresponding optimizer or task does not use. We use the ℓ_1 criterion for all regression heads. We use the margin-based sigmoid as our default symmetric loss. The pretrained checkpoints are `openai/whisper-base`, `google/bert_uncased_L-4_H-512_A-8`, and `ibm-research/MoLFormer-XL-both-10pct`.

Table 9: Shared per-task training protocol following RankUp's adopted USB standard setting (Wang et al., 2022; Huang et al., 2024). A dash marks a setting that does not apply.

	Vision (UTKFace)	Audio (BVCC)	Text (Yelp)	Chemistry (AqSolDB)
Backbone	Wide ResNet-28-2 (Zagoruyko & Komodakis, 2016)	Whisper-Base (Radford et al., 2023)	BERT-Small (Devlin et al., 2018)	MolFormer (Ross et al., 2022)
Pretrained	✗	✓	✓	✓
Training iterations	262,144	102,400	102,400	102,400
Evaluation interval	1,024	1,024	1,024	1,024
Learning-rate schedule	Cosine	Cosine	Cosine	Cosine
Warmup iterations	–	5,120	5,120	5,120
Labeled batch size	32	8	8	8
Optimizer	SGD	AdamW	AdamW	AdamW
Momentum	0.9	–	–	–
Learning rate	10^{-2}	2×10^{-6}	10^{-5}	10^{-5}
Weight decay	10^{-3}	2×10^{-5}	5×10^{-4}	5×10^{-4}
Layer decay	1.0	0.75	0.75	0.75
EMA decay	0.999	–	–	–
Regression criterion	ℓ_1	ℓ_1	ℓ_1	ℓ_1
Input representation	40×40 px	6.0 s at 16 kHz	512 tokens	512 tokens

The first three rows of Table 8 show the benchmark-specific values selected by Optuna (Akiba et al., 2019) with a Tree-structured Parzen Estimator (Bergstra et al., 2011). The AqSolDB configuration was fixed in advance and was not included in this search.

Chemistry follows the text setting, while the SymRankor-specific loss weights and batch ratios are fixed as shown in Table 8. The main regression term uses the ℓ_1 loss in Eq. (14). All other terms use a loss weight of 0.2 with $\beta = 1$. Each pseudo-ordinal class uses a batch size equal to that of the labeled batch.

The Gaussian vision ablation of Appendix I.1 comes from its own search and uses the values in Table 10. Every remaining setting follows the Vision column of Table 9.

Table 10: SymRankor-specific hyperparameters for the Gaussian vision ablation in Table 11.

Task	`arc_lb`	`arc_ulb`	`uratio`	`beta`
Vision (UTKFace)	0.75	3.40	19	4.80

I Additional Symmetric Losses

I.1 Gaussian Symmetric Loss on the Vision Task

Figure 4 plots the fraction of unlabeled pairs that clear the $\tau = 0.95$ gate. Five of the six runs open the gate within the first 10% of training. The Gaussian seed-2 run stays closed until 48.4%, so its unlabeled term acts over only half of the schedule. The $\tau = 0.95$ arms in Table 11 carry both the higher MAE and the wider seed spread.

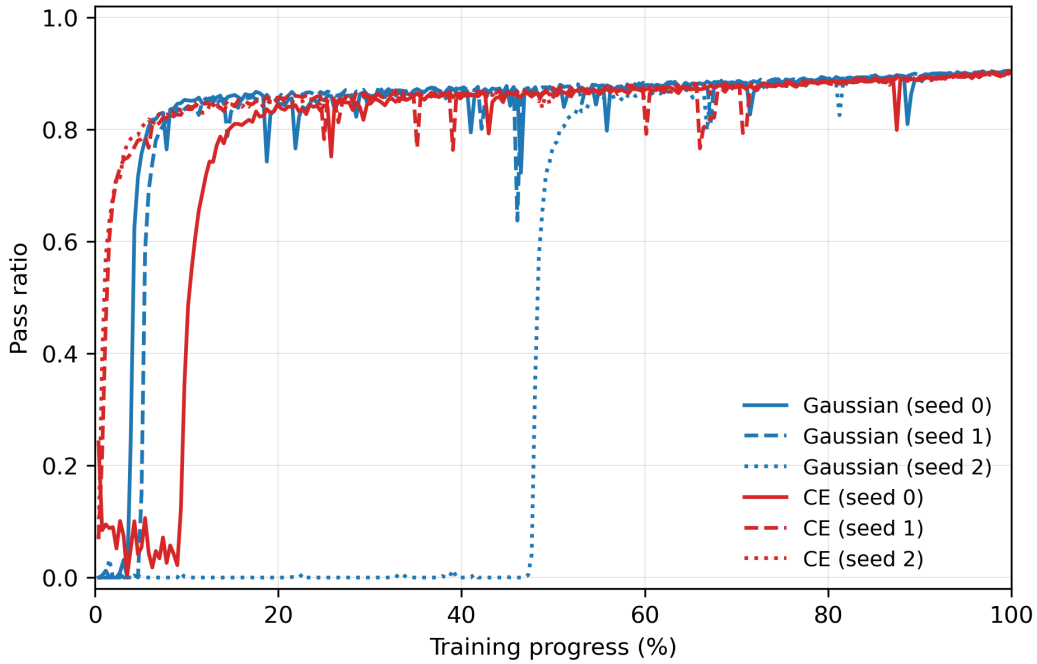


Figure 4: Fraction of unlabeled pairs passing the $\tau = 0.95$ confidence gate over training on the vision task, for the Gaussian and cross-entropy arms at three seeds. One Gaussian run admits no pair until 48.4% of training has elapsed.

Table 11: Ablation over the Gaussian symmetric loss and the confidence threshold τ on the vision task, in the format of Table 2. CE replaces the symmetric loss with the cross-entropy objective of RankUp. All four arms share the hyperparameters of Table 10. Results are reported as mean $\pm$ standard deviation over three runs ($n = 3$). Bold marks the best mean per metric.

Loss	τ	MAE $\downarrow$	R^2 $\uparrow$	SRCC $\uparrow$
Gaussian	0	**6.3950**$\pm$**0.0159**	**0.7971**$\pm$**0.0044**	**0.8528**$\pm$**0.0043**
CE	0	6.7258$\pm$0.1993	0.7791$\pm$0.0176	0.8487$\pm$0.0142
Gaussian	0.95	6.7644$\pm$0.5305	0.7713$\pm$0.0344	0.8430$\pm$0.0231
CE	0.95	6.9145$\pm$0.2992	0.7661$\pm$0.0202	0.8362$\pm$0.0118

Table 12: Per-seed results underlying Table 11. Time to 25% pass ratio is the training progress when the gate first admits 25% of unlabeled pairs. Final pass ratio is the share of unlabeled pairs admitted at the final step. At $\tau = 0$, every non-self pair is admitted from the first step. Bold marks the delayed Gaussian seed-2 run in Figure 4.

Loss	τ	Seed	MAE ↓	R^2 ↑	SRCC ↑	Time to 25% Pass Ratio (%)	Final Pass Ratio (%)
Gaussian	0	0	6.4070	0.7922	0.8530	0	100.00
		1	6.4009	0.7985	0.8484	0	100.00
		2	6.3770	0.8007	0.8569	0	100.00
CE	0	0	6.8606	0.7681	0.8402	0	100.00
		1	6.8199	0.7699	0.8407	0	100.00
		2	6.4968	0.7994	0.8651	0	100.00
Gaussian	0.95	0	6.3956	0.7919	0.8608	4	90.43
		1	6.5252	0.7904	0.8514	5	90.27
		2	**7.3723**	**0.7315**	**0.8169**	**48**	**89.96**
CE	0.95	0	7.1522	0.7514	0.8270	10	90.09
		1	6.5785	0.7891	0.8495	1	90.21
		2	7.0129	0.7577	0.8322	1	89.56

I.2 Results of the Additional Symmetric Losses

Under the default vision-task setting, Gaussian performs best among the additional symmetric losses. It achieves the lowest MAE and the highest R^2 and SRCC. The corresponding loss formulations are provided in Appendix B. The default setting follows RankUp in all configuration choices and uses $\beta = 1$.

Table 13: Performance of the additional symmetric losses on the vision task under the default experimental setting and $\tau = 0$. Results are reported as mean ± standard deviation over three runs ($n = 3$). Lower MAE and higher R^2 and SRCC indicate better performance.

Type	MAE ↓	R^2 ↑	SRCC ↑
Gaussian	**6.8495**±**0.0415**	**0.7620**±**0.0061**	**0.8379**±**0.0060**
Smooth ramp	6.9011±0.0775	0.7591±0.0071	0.8350±0.0059
Laplace	6.9193±0.2407	0.7575±0.0175	0.8330±0.0115
Softsign	7.0132±0.1701	0.7551±0.0088	0.8319±0.0058
Cauchy	7.1027±0.1155	0.7470±0.0115	0.8277±0.0089