Promoting Data Science

Volume 5 2025 Number 2

JOURNAL OF BEHAVIORAL DATA SCIENCE

ISBN: 2575-8306 (Print) 2574-1284 (Online)

https://jbds.isdsa.org

Journal of Behavioral Data Science V5N2 (2025)

https://isdsa.org

JOURNAL OF BEHAVIORAL DATA SCIENCE

No Publication Charge and Open Access

jbds@isdsa.org

List of Articles

Holly P. O'Rourke* and Chanler D. Hilley 1—27

A Guide to Specifying Effects in Latent Change Score Models with Moderated Mediation

Austin Wyman* and Zhiyong Zhang 28—42

Evaluating the Threat of Phantom Faces in Emotion Detection AI through Simulation

Shuo Xu, Hailiang Wang, Yijun Gao, Yixiang Li, and Meng-Ju Kuo* 43—63

More Than a Model: The Compounding Impact of Behavioral Ambiguity and Task Complexity on Hate Speech Detection

Journal of Behavioral Data Science, 2025, *5* (2), 1–27.
DOI: https://doi.org/10.35566/jbds/orourke

A Guide to Specifying Effects in Latent Change Score Models with Moderated Mediation

Holly P. O'Rourke[1][0000−0002−2927−0333] and Chanler D. Hilley[2][0000−0003−4766−9513]

[1] Department of Psychology, University of California at Riverside
holly.orourke@ucr.edu
[2] Department of Psychological Science, Kennesaw State University
chilley2@kennesaw.edu

Abstract. Latent change score (LCS) models are discrete-time longitudinal models that concurrently investigate growth over time and dynamic (lagged) relations among variables. Bivariate LCS models can be extended to multivariate scenarios with mediators and moderators, and mediation paths can be constrained or freely estimated across time. We provide a decision-making guide for model specification based on variable scale of measurement and hypothesized change processes. We then simulate two examples to illustrate how LCS models can be specified to estimate moderated mediation effects where the indirect effect from mediation is conditional upon values of the time-invariant moderator. We provide simulated data and annotated Mplus and R lavaan code.

Keywords: Latent Change · Mediation · Moderation · Moderated Mediation · Conditional Indirect Effects

1 Introduction

Many applied researchers conduct longitudinal studies with repeated measurements over time in order to refine their substantive theories with hypotheses about change processes. There are numerous available methods for longitudinal data analysis, and choosing the correct method for one's research question requires understanding the specific questions about change that a particular statistical model can answer. Methodological developments of latent change score (LCS) models (Cáncer & Estrada, 2023; Grimm, 2007; Grimm, An, McArdle, Zonderman, & Resnick, 2012; McArdle, 2009; McArdle & Grimm, 2010; O'Rourke, Fine, Grimm, & MacKinnon, 2022; Serang, Grimm, & Zhang, 2019; Usami, Hayes, & McArdle, 2016) have led to a recent wider adoption of applications of LCS models. Whereas latent growth curve (LGC) models answer questions about static change (i.e., one characterization of change over the entire course of the study), LCS models (the focus of this paper) answer questions about

dynamic change (i.e., lagged relations characterizing time-dependent changes over a study period) where future effects depend on past effects (Baltagi, 2021; Hsiao, 2014).

Much of the research on (and most of the applications of) LCS models to date have focused either on univariate models of change or on bivariate models with dynamic relations among two repeatedly measured variables. These models can be extended beyond the bivariate to incorporate additional variables which allow researchers to expand their causal theories of change. Recent work has examined the inclusion of mediators into LCS models (Goldsmith et al., 2018; Hilley & O'Rourke, 2022; Selig & Preacher, 2009; Simone & Lockhart, 2019), and in particular has focused on how to parameterize these models with respect to the traditional mediation literature stemming from the general linear model (Baron & Kenny, 1986; Judd & Kenny, 1981; MacKinnon, 2008). Researchers have also undertaken efforts to examine group differences of change in LCS models via inclusion of moderators (Cáncer, Estrada, & Ferrer, 2023; Estrada, Bunge, & Ferrer, 2023; Könen & Karbach, 2021; McArdle & Grimm, 2010; McArdle & Prindle, 2008).

Moderator and mediator variables can be related in several ways; one commonly specified relation is moderated mediation, where the indirect effect from mediation is conditional upon values of a moderator. Moderated mediation is indexed via a conditional indirect effect (CIE) (Hayes, 2018, 2022; Preacher, Rucker, & Hayes, 2007). To date, no work has been undertaken on model specification and interpretation of results from LCS models with both moderators and mediators. This paper uses illustrative examples to provide a guide to specifying CIEs in LCS models with variables that change dynamically over time, tying together prior simpler LCS model specifications to examine models that contains LCSs, mediators, and moderators (LCSMM models). We begin by providing definitions of moderators and mediators with details relevant to the ultimate LCSMM model of interest as well as an introduction to the LCS model. Next, we discuss some of the technical and practical details that must be considered when introducing CIEs into the LCS framework. We demonstrate the proposed methods with two simulated examples, each of which illustrates a different research question and corresponding parameterization of the LCSMM model.

1.1 Moderators

In modern behavioral research, group differences in bivariate relations are often of interest. Such relations can be examined via a moderator (Z) also known as an interaction effect, where Z influences the relation from X to Y. Interaction effects investigate whether the strength of the X to Y relation differs across varying levels of Z. Moderators, like any other predictor of Y, can take on any scale of measurement (though this paper focuses only on the use of binary moderators). A binary moderator can be added to a simple bivariate linear regression as shown in the following equations.

$$Y = i + d_1 X + e \tag{1}$$

$$Y = i + d_1X + d_2Z + d_3XZ + e \tag{2}$$

Equation 1 illustrates a bivariate regression equation where a predictor X is related to an outcome Y by a regression estimate d_1. In Equation 2, the moderator Z also predicts Y via a regression estimate d_2. The interaction XZ, which is the product of the predictors X and Z, also predicts Y in the model by way of the estimate d_3 (this estimate is the interaction effect). Re-arranging Equation 2 gives us an interaction term that can be used to predict how the relation of X to Y varies at values of Z, as shown in Equation 3:

$$Y = i + d_2Z + (d_1X + d_3Z)X + e \tag{3}$$

When Z is binary (e.g., coded 0 or 1), values of Z can be plugged in to Equation 3 to determine the effect of X on Y at different values of Z. For example, if Z = 0 the effect of X on Y would reduce to just d_1. If Z = 1, the effect of X on Y would be $d_1 + d_3$.

In research involving longitudinal change, moderators can be time-varying or time-invariant. A time-invariant moderator is a variable that does *not* change across time for individuals (e.g., random assignment to a treatment group), whereas a time-varying moderator is a Z variable that can vary across time for individuals (e.g., compliance with treatment protocol). Importantly, time invariance does not inherently imply that the effect of time-invariant Z on the X-Y relation cannot vary over the course of a study, only that values of the variable itself cannot vary over study duration.

1.2 Mediators

Mediators are another type of variable that can influence the relation of X to Y. In behavioral research, mediators represent theoretical mechanisms of change in terms of how X influences Y. Specifically, mediation analysis investigates whether the influence of X on Y is transmitted indirectly through an intervening variable (or mediator), M. Mediation makes the assumption that X temporally precedes M which in turn temporally precedes Y. For example, intervention researchers studying the effects of an intervention on a behavior may include a priori mediation theories about a mediator (or set of mediators) that the intervention (X) is designed to influence to ultimately lead to behavior change (Y); e.g., a smoking intervention (X) influences negative attitudes about smoking (M) which ultimately reduces smoking behaviors (Y). The equations below demonstrate the series of linear regression equations that capture these relations for a single mediator model (MacKinnon, 2008).

$$Y = i_1 + bM + c'X + e_1 \tag{4}$$

$$M = i_2 + aX + e_2 \tag{5}$$

In mediation analysis, the a path represents the influence of X on M in a regression equation predicting M. The b path represents the influence of M on Y in a separate regression equation predicting Y. The c' path refers to the influence

of X on Y while controlling for M, and is known as the direct effect. A single estimate of mediation can be quantified by taking the product of the a and b paths ab, known as the *indirect effect*, which represents the extent to which X influences Y through M (Alwin & Hauser, 1975).

Several formal statistical tests have been developed to assess the presence of mediation. The joint significance test (MacKinnon, Lockwood, Hoffman, West, & Sheets, 2002) is an offshoot of the causal steps approach (Baron & Kenny, 1986), which simultaneously assesses the significance of the a and b estimates to determine whether mediation is present. The joint significance test has been found to have the best balance of power and Type I error relative to other causal steps tests of mediation (MacKinnon et al., 2002). However, the joint significance test does not provide a test of significance for the overall estimate of the indirect effect, which is often preferred as it provides a single measure of mediation magnitude. The indirect effect ab can also be tested for significance by calculating a z test using a derived standard error (Sobel, 1982), or with asymmetric Monte Carlo bootstrapped confidence intervals of ab (MacKinnon, Fritz, Williams, & Lockwood, 2007; MacKinnon, Lockwood, & Williams, 2004). As these methods of assessing significance for mediation can be utilized for the different parameterizations of a and b in the LCS framework, both the joint significance test and bootstrapped confidence intervals of ab are used to assess significance of mediation throughout this paper.

Causal Assumptions of Mediation There are several assumptions about causality that are made in statistical mediation models, as causality is the defining feature that separates a mediation model from other models that are mathematically equivalent (for example, confounder models). The first assumption is *temporal precedence*, which is the assumption that X occurs temporally before M and M occurs temporally before Y within a given mediation model. Even if a mediation model contains only cross-sectional variables, temporal precedence assumes that at least a (very) small amount of time has elapsed between measurements of each subsequent variable in the mediational chain. This assumption is made in part to satisfy the second assumption described below.

Causal order is the second and related assumption, which relates to specification of the causal process among variables in a mediation model. Causal order assumes that the variables in the mediation model are ordered properly such that the causal process is correctly specified with X causing M, and M then causing Y. This second assumption clarifies the need for the first assumption; temporal precedence is a necessary but not sufficient requirement for establishing causality. Furthermore, an implication of this assumption is that there is no misspecification of causality. This encompasses a wide range of possible misspecifications such as no unmeasured confounders, no measurement error misspecification, and no backward causality (i.e., no reciprocal relations or reversed arrows in the mediation path model). In particular, handling the assumption of no unmeasured confounders often requires careful consideration of model specification and variable selection. When possible, the assumption of no unmeasured con-

founders is partially addressed by randomizing X (as in an intervention study), which addresses the assumption for the a path. However, this assumption of no unmeasured confounders typically cannot be not met for the b path, as M is characteristically a variable that cannot be randomized. The assumptions of causality can create unique challenges when adapting mediation models to frameworks beyond linear regression. These assumptions also have important implications for mediation models that specify CIEs, and each assumption has particularities to consider when conducting mediation analysis with longitudinal data (see the discussion section of this paper for a more in-depth treatment of these issues).

1.3 Conditional Indirect Effects

The condition under which mediation effects may differ for different groups (i.e., at different levels of a moderator) is sometimes referred to as "moderated mediation" (Hayes, 2015, 2018; Preacher et al., 2007). When the mediation a or b path is moderated such that the indirect effect is 1) not consistent across all individuals in a study and 2) systematically differs across subgroups in a sample, CIEs can be estimated to quantify the differences in mediation effects. Moderators can produce several different types of variations in the relations in a mediation model. A moderator Z can influence the a path such that the estimate of a (the effect of X on M) differs across levels of Z, which is referred to as first stage moderated mediation (Hayes, 2018). Alternatively, the moderator Z could influence the b path such that the estimate of b (the effect of M on Y) differs across levels of Z, a condition referred to as second stage moderated mediation (Hayes, 2018). First stage moderated mediation, which is the interaction relation utilized in our examples, is estimated in a linear regression framework with the following equations.

$$Y = i_1 + bM + c'X + e_1 \tag{6}$$

$$M = i_2 + a_X X + a_Z Z + a_{XZ} XZ + e_2 \tag{7}$$

Using this framework and building on the moderation equations above, CIEs can be calculated to provide group-specific indirect effects for a binary moderator Z. These group-specific CIEs are calculated using both the traditional mediation a path (denoted here as a_X) and the interaction term that quantifies the magnitude of variation in the relation of X to M at different values of Z (denoted here as a_{XZ}), along with the b path from the equation predicting Y. The CIEs are calculated as follows:

$$(a_X + a_{XZ} Z)b = a_X b + a_{XZ} bZ \tag{8}$$

For binary Z, values of Z can be entered into Equation 8 to calculate multiple CIEs that demonstrate group differences. For example, when Z = 0 the CIE would reduce to just $a_X b$ as in traditional mediation. When Z = 1, the CIE would be $a_X b + a_{XZ} b$.

In moderated mediation models, it is important to distinguish between auxiliary variables (i.e., Z) that moderate one or more of the mediation paths compared with interactions of the variables involved in the mediation processes; the distinction between the two is non-trivial. This paper focuses on cases where an auxiliary variable Z moderates the mediation a path. Traditional approaches to mediation have established the assumption that there is no XM interaction present, but it is possible that researchers could be interested in empirically testing whether X and M interact in their influence on Y (Valeri & VanderWeele, 2013).

The types of moderated mediation described above can be extended beyond linear regression to models with latent variables (Cheung, Cooper-Thomas, Lau, & Wang, 2021; Sardeshmukh & Vandenberg, 2017) or longitudinal models using a structural equation modeling (SEM) framework (Zhu, Sagherian, Wang, Nahm, & Friedmann, 2021). Much of the work in this space has focused on interactions among latent variables or moderators of static change (i.e., growth). In order to examine moderators of dynamic change over time in conjunction with mediation hypotheses, we now present details of the LCS framework.

1.4 LCS Models

LCS models are discrete-time longitudinal models that allow for investigation of both intraindividual change over two or more measurement occasions as well as interindividual differences in such change (Hamagami & McArdle, 2001; McArdle, 2001). LCS models capture measures of *both* static and dynamic (i.e., lagged) change over time. These models are fit using a SEM framework and are illustrated throughout the paper with path diagrams that utilize SEM path model notation.

The LCS framework was initially developed to overcome measurement error issues inherent in analyses with change scores of observed variables (Cronbach & Furby, 1970; Raykov, 1999). Using LCSs requires addressing the possibility of measurement error in a manner consistent with classical test theory such that for an observed variable y, the observed score for individual i at time t can be decomposed into a latent (true) score ly_{ti} and an error score e_{ti}:

$$y_{ti} = ly_{ti} + e_{ti} \tag{9}$$

The LCS framework defines latent scores as having fixed-unit autoregressive relations within a given variable over time, expressed as follows (Hamagami & McArdle, 2001; McArdle, 2001):

$$ly_{ti} = ly_{t-1i} + \Delta ly_{ti} \tag{10}$$

where a latent score for y at a given timepoint (ly_{ti}) is the sum of the latent score for y at the prior timepoint (ly_{t-1i}) and the change between latent scores for y from $t-1$ to t (Δly_{ti}). A univariate system of LCSs measuring change in a variable y over time includes two types of change, *constant change* and *proportional change*. Constant or static change where y_{a_i} is the additive (that is,

constant) change component for an individual i is functionally equivalent to the slope in a LGC model with no self-effect (Serang et al., 2019), and following notation from Cáncer, Estrada, Ollero, and Ferrer (2021) has a mean μ_{y_a}, variance $\sigma^2_{y_a}$, and covariance with the initial level σ_{y_0,y_a} and is referred to as an additive component throughout this paper.

The additive component models interindividual rates of change while maintaining a constant rate of intraindividual change. Proportional change β_y relates prior latent level at $t-1$ to later latent change between t–1 and t. In LCS models, proportional change allows prior latent levels to influence later latent change in a given variable. These change parameters can be combined into a single dual change model, which includes both additive and proportional change components.

$$\Delta ly_{ti} = y_{ai} + \beta_y ly_{t-1i} \tag{11}$$

Together, the additive and proportional change components describe an exponential trajectory in the dual change model. The dual change model described here is not the only possible specification of univariate models for latent change. One could include only the additive component, which would result in a static model equivalent to a LGC model that is defined in terms of latent change, or only the proportional change β_y as a model of self-effect. This univariate LCS system can also be extended to study longitudinal bivariate relations. The relation between two longitudinal variables in the LCS framework is most often measured using coupling, the influence of prior latent levels of one variable on later latent change in another. The interpretation of the coupling parameter γ_{yx} resembles the proportional change described earlier. However, coupling is a time-dependent effect that provides an estimate of the extent to which the preceding measurement's level of one variable impacts the trajectory of change in *another* variable at a subsequent measurement occasion. Specifically, significant coupling from a variable x to a second variable y means that x is a *leading indicator* of scores on y. We can introduce a coupling parameter into the dual change model such that x is a leading indicator of y:

$$\Delta ly_{ti} = y_{ai} + \beta_y ly_{t-1i} + \gamma_{yx} lx_{t-1i} \tag{12}$$

We build on this bivariate equation by adding mediators and moderators in the coming sections.

1.5 Conditional Indirect Effects in LCS Models

To date, there has been no methodological treatment of first stage moderated mediation in LCS models. Much of the methodological literature on mediation in the LCS framework only examines latent change between two timepoints for any given variable (Goldsmith et al., 2018; Selig & Preacher, 2009; Simone & Lockhart, 2019). More recent work has examined how to specify LCS models to include either cross-sectional or longitudinal mediators with more than two timepoints (Hilley & O'Rourke, 2022). With respect to research on moderators or

group differences in change in the LCS framework, moderators have typically not been the primary focus of methodological investigations and examinations have been limited to moderators of univariate change (i.e., univariate cohort effects) (Cáncer et al., 2023; Estrada et al., 2023; Könen & Karbach, 2021); applied demonstrations of moderation in bivariate LCS models exist but are rare (see McArdle & Grimm, 2010). However, despite the lack of methodological guidance, several applied studies have included time-invariant binary moderators in LCS models (Gradinger, Yanagida, Strohmeier, & Spiel, 2015; Griffiths, Kievit, & Norbury, 2022; Zaccoletti et al., 2020). Given the obvious interest in applications of group differences in change in the LCS framework, it is natural to extend the framework to examine group differences in change with respect to mediation paths.

In this paper, we present two models that illustrate different ways to specify CIEs for first stage moderated mediation in the LCS framework. We begin by defining a general LCSMM model with a time-invariant binary moderator. From this general model, we then combine the information regarding CIEs and LCSs that is provided in the introduction to demonstrate multiple ways to specify CIEs in LCSMM models. All computer code and supplementary material for this tutorial can be found on GitHub at `https://github.com/horourke/CIE_LCS`.

2 Example Model Setup

To demonstrate the different specifications of CIEs in LCSMM models, we begin with a keystone LCS model. In this model, X_1 (X at $t = 1$) is a continuous variable that is measured at a single timepoint and is assumed to occur first temporally in the model. M and Y are continuous variables that are measured repeatedly at five timepoints, each with a univariate dual change structure. The a path for mediation is specified such that X predicts the additive component of M. This is a more parsimonious way of specifying a, in contrast to a model where X has multiple constrained paths predicting latent change in M (Hilley & O'Rourke, 2022). The mediation b path is specified by a coupling relation from M to Y, where prior latent level of M influences later latent change in Y. Coupling paths are lagged such that latent levels of M at t–1 predict later latent change in Y (between t–1 and t) to meet the mediation assumption of temporal precedence. A measurement schedule that begins at the timepoint directly after measurement of X is assumed for both M and Y. X also predicts the additive component of Y (akin to a mediation c' direct effect). The moderator Z is a binary variable that is time-invariant (i.e., the onset of the group difference is assumed to occur before the measurement of all other variables in the model and to persist without change for the duration of the study). The moderator in the keystone model influences the a path such that there are group differences across Z in the influence of X on the additive component of M (i.e., a first stage moderated mediation model).

This paper demonstrates estimating CIEs when the moderator Z and the product of X and Z are included as time-invariant covariates. The time-invariant

covariate method uses an extension of the moderator equations described in the introduction and incorporates the moderator directly into the statistical model. In this approach, an interaction term is estimated as a parameter relating the product of X and Z to the additive component of M (i.e., the interaction is first stage moderated mediation) and is included in the equation of each indirect effect. Values of Z are substituted into the resultant equation to calculate CIEs at each value of Z. We provide code in *Mplus* and R to estimate these models and calculate the resultant CIEs at values of binary Z.

These models may also be estimated using an equivalent multiple group approach. The multiple group approach was developed to investigate group differences in change (McArdle & Hamagami, 1996) and was originally utilized as a way of examining group differences for factor structures (Jöreskog, 1971). With this approach, an invariance method is used where each parameter in the model is explicitly specified to be either constrained to be equal across groups or to vary across groups. This approach allows researchers to examine group differences in any parameter of the model while defaulting to constraining all other parameters that are not explicitly freed to vary across groups. When a moderator is binary and time-invariant, the multiple group approach to fitting a LCSMM model produces two sets of model estimates, one for each group on the moderator. The a and b paths can be freely estimated across groups, with two CIEs calculated individually as $a^{(Z=0)}b$ and $a^{(Z=1)}b$ using the MODEL CONSTRAINT command in *Mplus*. A Wald χ^2 test can then used to assess whether the two CIEs significantly differ from one another, which would indicate that the a path of X to M (and therefore the entire indirect effect ab) was moderated by Z. Although not discussed in detail in this article, code for this approach is also provided on our GitHub.

Given that most models in methodological work on LCS models present constrained coupling paths, it is correspondingly rare to find applications where LCS models are fit with freely estimated coupling paths in empirical studies. However, varying the typical constraints on coupling may be necessary for certain research questions about change, and doing so has implications for how CIEs are calculated in LCSMM models (as we will see in the examples). The traditional use of a coupling path that is constrained across time resulting in a single estimate would result in only one CIE per value of the moderator. However, specifying a LCSMM model with coupling paths that are freely estimated across time results in multiple estimates of b, and thus requires that for each value of the moderator we calculate multiple CIEs (one per estimate of b). Therefore, we present two examples in this paper: one example with the most common specification where b is a coupling parameter constrained to be equal across time, and one example where there are multiple b paths that are coupling parameters freely estimated at each wave.

2.1 Data Generation and Analysis

For each example, an illustrative data set was simulated in R (R Core Team, 2020) using the keystone model as a baseline and varying constraints on the b

path. Parameter values were selected to produce data trajectories typically seen in longitudinal behavioral research, and were varied such that the a paths were equal in magnitude but had opposing signs across groups on Z. When data were simulated with a b path that varied across time, the b path magnitude decreased at each successive wave. Our rationale for using this pattern of variation for the magnitudes comes from the mediation literature (O'Rourke & MacKinnon, 2015, 2018), where Y outcomes that are more distal (i.e., farther away in time of measurement) typically have weaker relations to the mediator than outcomes that are more proximal (i.e., closer in time of measurement). Correlations among latent initial score and additive change component means were fixed to .5 (large) based on correlations commonly observed in applications of LCS models (Grimm, 2007). Population parameters used to simulate the data are shown in Table 1. In both of the following examples, estimates from the models that were fit to the simulated data were generally unbiased.

Table 1. Parameter Population Values for Simulated Examples

	Constrained	Freely Estimated
Parameter	Example 1	Example 2
μ_X	0	0
σ^2_X	1	1
Univariate M		
μ_{m0}	6	6
μ_{ma}	0.9	0.9
β_m	-0.05	-0.05
σ^2_{m0}	0.49	0.49
σ^2_{ma}	0.01	0.01
$\sigma^2_{e(m)}$	0.025	0.025
Univariate Y		
μ_{y0}	7	7
μ_{ya}	-0.5	-0.5
β_y	0.1	0.1
σ^2_{y0}	0.36	0.36
σ^2_{ya}	0.04	0.04
$\sigma^2_{e(y)}$	0.101	0.101
Mediation/Moderation		
a_{X_1}	-0.4	-0.4
a_Z	0	0
a_{X_1Z}	0.8	0.8
b	-0.13	-
b_3	-	-0.23
b_4	-	-0.18
b_5	-	-0.13
b_6	-	-0.08
c'	0.1	0.1

After data simulation, models were then fit to the data using *Mplus* (Muthén & Muthén, 2017) and the lavaan package in R (R Core Team, 2020; Rosseel, 2012). Data generation code (R), analytic code (*Mplus* and R), and simulated datasets for each example can be found on GitHub.

2.2 Power Analyses

For the datasets in each example, a sample size of 520 ($n = 260$ per moderator group) was chosen based on recent simulation work recommending minimum sample sizes for mediation with LCSs (Simone & Lockhart, 2019). Although in the traditional regression framework, including moderators often results in lower power to detect effects, the sample sizes provided in Simone and Lockhart (2019) allowed for detection of significant indirect effects at magnitudes that would reasonably be seen in empirical research. Beyond this simulation study, there is little empirical work on sample size and power for mediation in the LCS framework. For readers interested in conducting power analyses for the models described in this paper, we have provided the code for a user-friendly method to conduct power analyses for both models described below. Because LCS models in general have complex model specifications that require several parameters to be fixed to 0 or 1, Monte Carlo simulations are not straightforward in many existing R packages; instead, we are using a 2-step process to conducting power in *Mplus* (Muthén & Muthén, 2017).

First, we conducted the LCSMM analyses described in this paper and used the SAVEDATA: ESTIMATES = function in *Mplus* to save the parameter estimates as a data file to be used as population parameters in the next step. This first step requires the use of a "real" data file; we used the data we simulated in R in this step. Additionally, MODEL CONSTRAINT statements to compute the CIEs are not included in the code for the first step, as they are not relevant for generating the data file containing population parameters. In the second step, we conducted a power analysis using the Monte Carlo procedure in *Mplus*. Unlike typical *Mplus* Monte Carlo power analyses in which the population parameters are set in the MODEL POPULATION statement, the population values come from the data file saved in the first step. In the second step, a MODEL CONSTRAINT statement was used to compute the CIEs. Using this process, power was examined for all estimates in the LCSMM models, including the CIEs.

3 Example 1: Constrained Dynamic Paths

In example 1, we demonstrate how to specify and estimate CIEs for a LCSMM model with constrained coupling for the b path[1].

[1] In Figures 1 and 2, paths sharing labels are constrained to be equal. Unlabeled paths are constrained to be 1, except the paths marked by an asterisk which are freely estimated. Variances and covariances are represented with double-headed arrows. Variances and covariances not shown are constrained to be 0, except covariances among the residuals for observed M and Y which are excluded from this path model for visual parsimony.

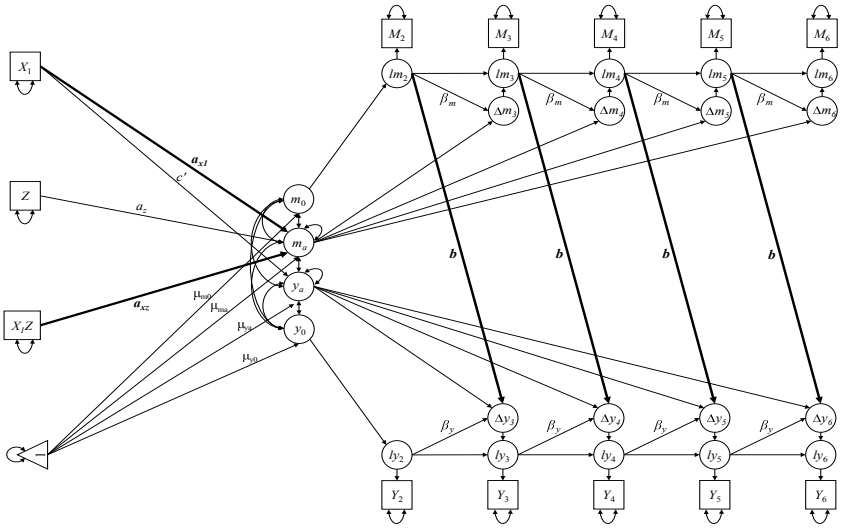

Figure 1. LCSMM Model with Constrained Coupling for b

The following equations express this LCSMM model and correspond to the path diagram in Figure 1.

$$\Delta lm_{ti} = m_{ai} + \beta_m lm_{t-1i} \tag{13}$$

$$m_{ai} = a_{X_1} X_1 + a_Z Z + a_{X_1 Z} X_1 Z \tag{14}$$

$$\Delta ly_{ti} = y_{ai} + \beta_y ly_{t-1i} + blm_{t-1i} \tag{15}$$

$$y_{ai} = c' X_1 \tag{16}$$

In Equation 14, the mediation a path is distinguished from the other parameters in the equation with a subscript, a_{X_1}, to denote that it is the path relating X_1 to the additive component of M. An interaction is formed by taking the product of X_1 and Z, and this product is included in the equation as a third predictor of the additive component of M by way of the parameter $a_{X_1 Z}$ (the interaction term). The main effect of Z predicting M is also included, as represented by the parameter a_Z.

The CIEs are estimated using the following equation.

$$ab_{CIE} = (a_{X_1} b) + (a_{X_1 Z} b) Z \tag{17}$$

This equation is quite similar to the CIEs introduced in Equation 8 from a regression framework, although in this example b is a longitudinally constrained coupling parameter rather than a linear regression path. To calculate CIEs from model estimates, Equation 17 is programmed directly into the analysis script in *Mplus* and R using MODEL CONSTRAINT statements. We use Wald tests of significance (Wald, 1943) to assess the difference in the CIEs, therefore providing evidence of significant, stage one moderated mediation. The Wald test null hypothesis was specified as

$$a^{(Z=0)}b - a^{(Z=1)}b = 0 \tag{18}$$

Bootstrapping was also conducted to generate bootstrapped confidence intervals of the respective CIEs and to determine their significance in terms of inclusion of 0[1].

3.1 Conditional Indirect Effect Interpretations

Table 2 contains results from the simulated example for which we provide interpretation of the relevant estimates used in calculating CIEs. The mediation a path was negative and significant ($a_{X_1} = -0.373$), indicating that for a one-unit increase in X_1, the additive component of M decreased by .373 units. The coupling b path was also negative and significant ($b = -0.146$), indicating that higher values of M were a leading predictor of lower values on Y. Using the joint significance test, significance of the a and b paths resulted in a conclusion that overall mediation was present (without consideration of the moderator), such that higher values of X predicted a more negative additive component of M and higher prior values of M predicted smaller changes in Y at each subsequent wave.

$$ab_{CIE} = (-0.373 * -0.146) + (0.768 * -0.146)Z = -.054 - 0.112Z \tag{19}$$

The equation above resulted in CIE estimates of $a^{(Z=0)}b = 0.054$ and $a^{(Z=1)}b = -0.057$. Bootstrapped confidence intervals of the CIE estimates did not include zero, indicating that mediation was present at both values of the moderator.

The Wald test of equality supported results with respect to significance of the moderation estimate, ($\chi^2(1, N = 520) = 106.830, p < .001$), providing evidence that moderation of the a path across values of Z resulted in significant group differences in the CIEs. Considering all results from this model, we can conclude that mediation was significant for both groups, and that the impact of X on the additive component of M differed between the groups, resulting in mediation CIEs that were both significant but with opposite signs (and significantly different from one another). Additionally, the power analyses described previously demonstrated power approaching 1 for both CIEs.

[1] The MODEL TEST command in *Mplus* cannot be used in conjunction with bootstrapping, so if bootstrapped confidence intervals of the CIEs are desired, the Wald test needs to be conducted in a separate *Mplus* script.

Table 2. Unstandardized Estimates from LCSMM Model, Constrained Coupling

Parameter	Estimate (SE)	Lower 95% CI	Upper 95% CI
Univariate M			
μ_{m0}	6.017 (0.04)***	5.937	6.097
μ_{ma}	0.833 (0.037)***	0.761	0.904
β_m	-0.039 (0.005)***	-0.050	-0.029
σ^2_{m0}	0.466 (0.031)***	0.404	0.527
σ^2_{ma}	0.009 (0.001)***	0.008	0.011
$\sigma^2_{e(m)}$	0.025 (0.001)***	0.023	0.027
Univariate Y			
μ_{y0}	6.964 (0.046)***	6.874	7.055
μ_{ya}	-0.346 (0.175)*	-0.683	0.002
β_y	0.094 (0.015)***	0.065	0.123
σ^2_{y0}	0.401 (0.028)***	0.344	0.455
σ^2_{ya}	0.045 (0.007)***	0.033	0.059
$\sigma^2_{e(y)}$	0.105 (0.004)***	0.097	0.112
Mediation/Moderation			
c'	0.115 (0.012)***	0.092	0.138
a_{X_1}	-0.373 (0.008)***	-0.389	-0.357
a_Z	-0.006 (0.008)	-0.022	0.011
a_{X_1Z}	0.768 (0.011)***	0.748	0.789
b	-0.146 (0.015)***	-0.175	-0.117
$a^{Z=0}b$	0.054 (0.005)***	0.044	0.065
$a^{Z=1}b$	-0.057 (0.006)***	-0.069	-0.047

*$p < .05$, **$p < .01$, ***$p < .001$.

4 Example 2: Freely Estimated Dynamic Paths

In our second example, the b coupling paths were freely estimated across time. The path model for this example is shown in Figure 2. The equation predicting the LCSs for Y demonstrates how the model in this example differs from the model in example 1:

$$\Delta ly_{ti} = y_{ai} + \beta_y ly_{t-1i} + b_t lm_{t-1i} \tag{20}$$

In this equation, the b path is now time-dependent as denoted by the subscript t such that coupling is freely estimated across timepoints. The estimation of multiple b paths necessitated the calculation of multiple CIEs for a given value of Z, as there were four LCSs and thus four corresponding b paths. With two values of Z, eight CIEs were calculated, four each for $a^{Z=0}b_t$ and $a^{Z=1}b_t$. Four equalities of CIEs were specified in the Wald Test such that the CIEs were equal with time held constant, resulting in the following null hypothesis for the Wald test.

$$a^{(Z=0)}b_t - a^{(Z=1)}b_t = 0 \tag{21}$$

Although four equalities were specified for the Wald test (one for each timepoint), only one χ^2 estimate was provided for all of the tests. Therefore, in this

example the Wald test provided information on whether at least one of the sets of CIEs differed between groups, considering all timepoints. Bootstrapping was also conducted to produce confidence intervals of the individual estimates as well as the eight CIEs.

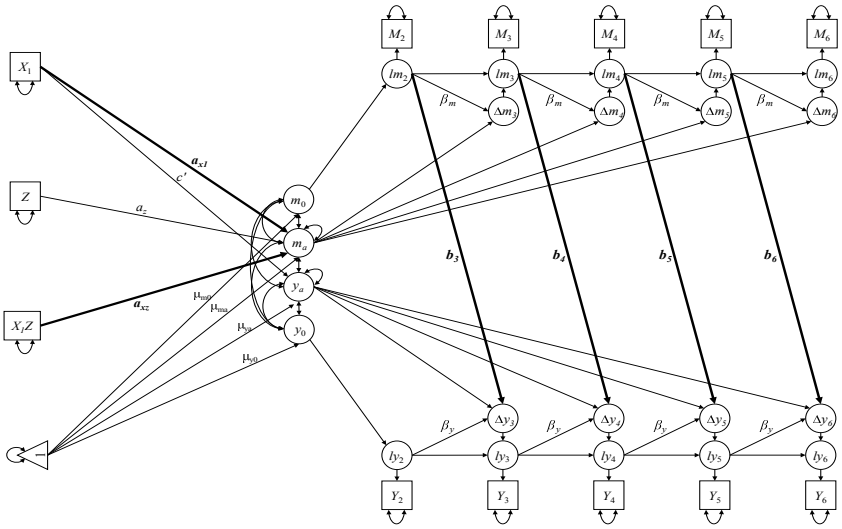

Figure 2. LCSMM Model with Freely Estimated Coupling for b

4.1 Conditional Indirect Effect Interpretations

Table 3 shows model results from this example's estimation. As with example 1, the mediation a path was negative and significant ($a_{X_1} = -0.396$) such that a one-unit increase in X_1 resulted in a 0.396-unit decrease in the additive component of M. All b path estimates were negative and significant ($b_3 = -0.230$, $b_4 = -0.174$, $b_5 = -0.127$, and $b_6 = -0.072$) where with each subsequent wave, the magnitude of b weakened. As indicated by the trend toward zero in the b paths, higher prior levels of M were associated with smaller subsequent changes in Y, however this relation weakened over the course of the study. Both the$a$ path and all freely estimated b paths were statistically significant, supporting evidence that mediation was present across all waves in accordance with the joint significance test. The conclusions from these results can be interpreted such that higher values of X predicted lower values of M over time, and subsequently higher prior values of M predicted smaller changes in Y, with the prediction of M on change in Y weakening over time.

Table 3. Unstandardized Estimates from LCSMM Model, Freely Estimated Coupling

Parameter	Estimate (SE)	Lower 95% CI	Upper 95% CI
Univariate M			
μ_{m0}	5.966 (0.04)***	5.886	6.044
μ_{ma}	0.905 (0.037)***	0.831	0.977
β_m	-0.051 (0.005)***	-0.061	-0.041
σ^2_{m0}	0.465 (0.03)***	0.404	0.522
σ^2_{ma}	0.009 (0.001)***	0.007	0.011
$\sigma^2_{e(m)}$	0.024 (0.001)***	0.022	0.026
Univariate Y			
μ_{y0}	6.95 (0.042)***	6.869	7.032
μ_{ya}	-0.614 (0.201)**	-1.019	-0.231
β_y	0.116 (0.035)**	0.051	0.187
σ^2_{y0}	0.381 (0.027)***	0.326	0.432
σ^2_{ya}	0.034 (0.007)***	0.022	0.05
$\sigma^2_{e(y)}$	0.101 (0.004)***	0.094	0.108
Mediation/Moderation			
c'	0.105 (0.011)***	0.085	0.127
a_{X_1}	-0.396 (0.007)***	-0.41	-0.382
a_Z	0.004 (0.009)	-0.013	0.022
a_{X_1Z}	0.799 (0.011)***	0.777	0.82
b_3	-0.23 (0.019)***	-0.267	-0.192
b_4	-0.174 (0.015)***	-0.204	-0.145
b_5	-0.127 (0.013)***	-0.154	-0.101
b_6	-0.072 (0.014)***	-0.1	-0.046
$a^{Z=0}b_3$	0.091 (0.007)***	0.076	0.106
$a^{Z=0}b_4$	0.069 (0.006)***	0.058	0.08
$a^{Z=0}b_5$	0.05 (0.005)***	0.04	0.061
$a^{Z=0}b_6$	0.029 (0.005)***	0.018	0.039
$a^{Z=1}b_3$	-0.093 (0.008)***	-0.107	-0.078
$a^{Z=1}b_4$	-0.07 (0.006)***	-0.082	-0.059
$a^{Z=1}b_5$	-0.051 (0.005)***	-0.062	-0.041
$a^{Z=1}b_6$	-0.029 (0.005)***	-0.04	-0.019

*$p < .05$, **$p < .01$, ***$p < .001$.

The interaction term representing the product of X and Z predicting the additive component of M was positive and significant ($a_{X_1Z} = 0.799$), indicating that there were significant group differences on Z in the prediction of X on the additive component of M. The influence of Z on the additive component of M (main effect) was not significant. The CIEs were estimated by utilizing the model estimates in the following equations:

$$ab_{CIE_3} = (-0.396 * -0.230) + (0.799 * -0.230)Z \tag{22}$$

$$ab_{CIE_4} = (-0.396 * -0.174) + (0.799 * -0.174)Z \tag{23}$$

$$ab_{CIE_5} = (-0.396 * -0.127) + (0.799 * -0.127)Z \tag{24}$$

$$ab_{CIE_6} = (-0.396 * -0.072) + (0.799 * -0.072)Z \tag{25}$$

Bootstrapped confidence intervals for the CIEs indicated that mediation was significant at all timepoints for both values of the moderator, as each of the eight confidence intervals did not include zero. With respect to group differences of the CIEs, the Wald test was significant, $\chi^2(4, N = 520) = 200.045, p < .001$, indicating that at least one of the sets of CIEs at a given timepoint differed between groups on the moderator. Additionally, Z moderated the influence of X on M such that the a_{X_1} path varied across values of Z which resulted in intergroup CIEs of opposing signs, with at least one timepoint having statistically significant group differences in indirect effects. As the Wald test does not give us specific information on which of the pairs of CIEs differed significantly at each time point, plotting the CIEs is a useful way to interpret the moderated mediation effect from the LCSMM model. Figure 3 contains a plot of the CIEs across values of b at each timepoint grouped at each value of Z for example 2.

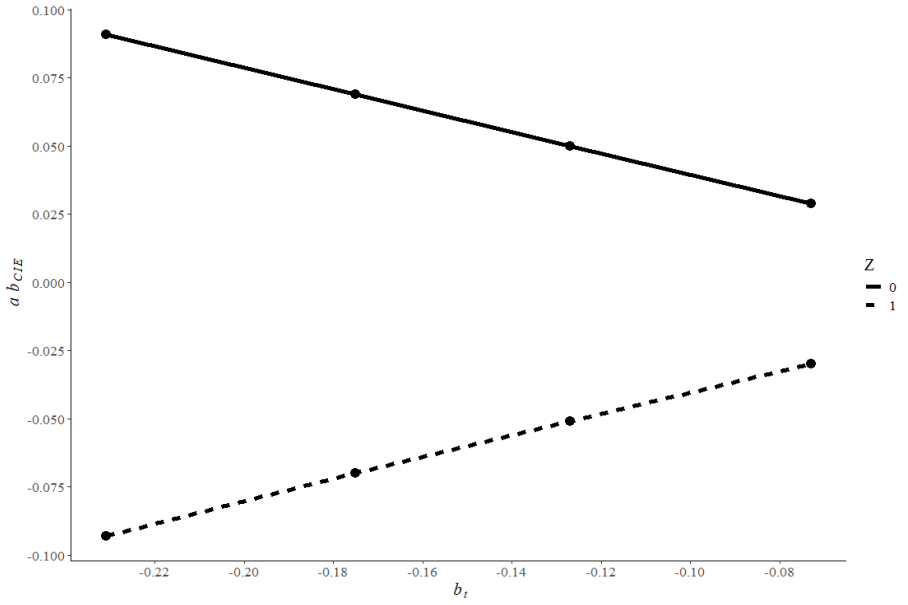

Figure 3. Comparison of CIEs Across Groups for LCSMM Model with Freely Estimated Coupling for b

This plot shows that as the estimate of b decreased over time, the difference in the CIEs between values of Z also decreased such that at the final time point the CIEs had the smallest difference. Code for reproducing this plot in R can be found on GitHub. Additionally, power analyses demonstrated that power approached 1 for all eight estimates of the CIEs.

5 Discussion

In this paper, we demonstrated an approach for specifying CIEs in LCSMM models which can be used to address research questions about group differences in mediation models of dynamic change over time. We illustrated two specifications of LCSMM models to show how binary group differences in indirect effects can be examined within the LCS framework using a time invariant covariate approach with a binary moderator when the mediation b paths are either freely estimated or constrained to be equal across time. As the LCS framework continues to grow in popularity among applied researchers, the models described in this paper will be useful for refining theories of behavior change, particularly with respect to group differences in mechanisms of change.

5.1 Mediation Considerations

The models presented in this paper conceptualize the mediation paths as those that predict additive components (in the case of the a path) and as coupling parameters (in the case of the b and c' paths), all of which involve a type of change as the outcome. Although these specifications are in line with prior work on LCS models with mediators (Hilley & O'Rourke, 2022), the mediation paths could be specified differently with respect to change (e.g., a as the influence of X on latent levels of M; b as the influence of prior latent levels of M on later latent levels of Y; b as the influence of prior latent change in M on later latent change in Y; etc.). Additionally, by specifying the a path as $X \rightarrow M_a$, the models we have presented are also more parsimonious than other options, like $X \rightarrow \Delta_{mt}$. The alternative specification may be more appropriate for certain research scenarios, such as if the research question hypothesized that X would predict change in M differentially over time (i.e., when the influence of X on change in M differs over time).

Additionally, for each example the X variable was assumed to be a continuous normally distributed variable. Quite often, researchers conducting studies with hypothesized mediators measure X as a categorical randomized variable in an attempt to address the no unmeasured confounders assumption of mediation for the a path. The models presented here could easily be extended to include interaction terms between categorical X and Z; for example, if X and Z are binary, the product XZ could be coded as a 4-category variable. Goldsmith et al. (2018) describe a modified LCS mediation model with a randomized, 4-group variable for X (although CIEs were not specified for these models).

Causality Considerations As described in the introduction, the causality assumptions inherent to mediation are what separate mediation from other types of three-variable models, and this is true for mediation in any framework. We now address several considerations that are specific to causality for mediation in the LCS framework.

Temporal Precedence. The LCS framework provides both opportunities and challenges for examining mediators. To begin with the benefits, mediation in LCS models can allow researchers to examine effects that are known to be lagged and therefore temporal precedence is known to be met (e.g., $X_1 \rightarrow \Delta_{m_2-m_3} \rightarrow \Delta_{y_3-y_4}$), or a corresponding coupling specification). This is an advantage over mediation models with cross-sectional data where X, M, and Y are all measured at the same time (e.g., $X_1 \rightarrow M_1 \rightarrow Y_1$). However, researchers must be deliberate in how they specify their indirect effects to match theories of change, as specifying change-change paths for mediation can result in measurements of contemporaneous change (e.g., $\Delta_{m_2-m_3} \rightarrow \Delta_{y_2-y_3}$) that do not satisfy the temporal precedence assumption. Circumstances do exist where it is appropriate to specify contemporaneous change paths, such as if variables are measured at the same timepoint but prior research indicates an underlying causal process that occurs at a faster rate than the measurement timeline can capture (Goldsmith et al., 2018; Hilley & O'Rourke, 2022). The theory of change should always be considered when determining the specification of change for longitudinal mediation models.

Potential Confounders of Mediation. The assumption of no potential confounding influences with respect to establishing causality in mediation is often both the most problematic in terms of influencing model results when the assumption is violated, and the most difficult to address. When X is not randomized (and as we have mentioned previously, M is very often not randomized), bias is introduced into the estimates of each of the mediation paths where M and Y are being influenced by unmeasured confounders. Some longitudinal mediation frameworks (including the LCS framework) can partially address the issue of unmeasured confounders influencing M and Y by including correlated measurement errors between M and Y across timepoints, and the current recommendation is to use contemporaneous residual covariances among M and Y at each measurement of t (Goldsmith et al., 2018). Although they are not shown in our path diagrams, we have utilized this method to address potential confounding in our examples; all of the LCSMM models presented here include correlations between residuals for M and Y at each timepoint. This method addresses the assumption of no unmeasured confounders for the b path, but unless X is randomized, the assumption is not met for the a path. When X is a cross-sectional observed variable in a LCSMM model, it is recommended to use methods for dealing with potential confounders that were developed for mediation models in the linear regression framework (Hilley & O'Rourke, 2022; MacKinnon & Pirlott, 2015).

5.2 Moderation Considerations

We now turn to some considerations for the moderation portion of the LCSMM model and calculations of the CIEs. In LCS models, coupling represents the extent to which prior levels of one variable influence later change in another. Thus, with the specification utilized in this paper, the mediation b path is represented by coupling between M and Y. In both examples, the moderator was a binary,

time-invariant variable influencing the mediation a path. However, as described below, the LCSMM models presented in this paper can be extended to capture moderation of the b path or moderators that are time-varying or continuous, although additional methodological research is needed in these areas.

Continuous vs. Binary Moderators In this paper we demonstrated approaches for calculating CIEs in LCSMM models using only binary, time-invariant moderators. It is important to note that if the moderator was continuous and time-invariant, the covariate and multiple group approaches to estimation would not give equivalent CIEs due to the adjustments that would have to be made to the calculations for each approach. In the multiple group approach, prior to estimation synthetic categorical groups would have to be created by categorizing the continuous Z variable. Assuming Z is normally distributed, this would typically be done by using a "high/medium/low" binning schema separating the data into thirds and then running a multiple group analysis for the binned groups. However, this approach is not recommended in practice due to the loss of variability stemming from binning a continuous variable (Altman & Royston, 2006).

In contrast, for the time-invariant covariate approach, the product XZ could be computed and included in analysis in the same manner as it was in the examples presented here. Researchers would then choose "high/medium/low" values of Z for which to calculate CIEs (typically $-1SD/M/+1SD$ when Z is normally distributed). The time-invariant approach thus would retain all of the original variability from the continuous moderator by including it in the model as a product with X. By contrast, the multiple group approach estimates would come from "groups" that are in reality just separated groups of the same sample. Thus, calculating CIEs from mediation estimates using the multiple group approach would result in different CIE estimates as compared to using the covariate approach. This difference in estimates would be even more pronounced if the continuous time-invariant moderator was not perfectly normally distributed.

XM Interactions in LCSMM Models In the introduction of this paper we described two types of interactions involving mediation, first and second stage moderated mediation. Each of these types of moderation involves interaction with the predictor X and a moderator Z, and our examples throughout the paper used only first stage moderated mediation models. However, a single mediator model can be extended to include interactions between X and the mediator M, with no additional auxiliary variables in the model (i.e., XM interactions). The models we described here could be extended to include XM interactions as well. Methods have recently been developed for estimating XM interactions with latent variables using a causal (rather than traditional regression) framework (Gonzalez & Valente, 2023), but the causal effects can easily be converted to traditional CIEs for XM interactions (MacKinnon, Valente, & Gonzalez, 2020).

5.3 Constraint Considerations

Next, we describe a more conceptual consideration, which is how to choose whether to constrain vs. freely estimate the b path (or other dynamic change paths) in LCSMM models. The choice should ultimately depend on the researcher's hypotheses about change over time. Given a single research question, it may sometimes make sense to either constrain or freely estimate dynamic change paths based on the given timeline of a research study. As an example, suppose we have a developmental theory where we expect to observe coupling between social preference and antisocial behavior in adolescence such that social preference is a negative leading predictor of adolescent antisocial behavior (Buil, Van Lier, Brendgen, Koot, & Vitaro, 2017). A study that occurs over a one-year period during adolescence might hypothesize that coupling between social preference and antisocial behavior is consistent over multiple measurements in that relatively short study period, and thus constraining the coupling parameter to be equal across timepoints will both result in best model fit and be consistent with the theory. However, if the same study were conducted over a longer period starting in early adolescence and across the transition to adulthood (when salience of social preference theoretically increases), the best representation of the developmental theory would be to freely estimate the coupling paths such that they are allowed to strengthen over time.

Although the selection of framework for longitudinal data analysis should be driven by the research question at hand, the examples provided in this paper highlight some of the benefits and challenges of each type of model. For example, when the coupling paths are freely estimated, researchers will obtain CIEs for each of the estimated paths and each group of the moderator (e.g., in our examples where two levels of the moderator and four coupling paths from M to Y resulted in eight CIE estimates). When these coupling paths do truly differ over time, models that constrain them to be equal would be misspecified. However, interpretation of the CIEs may be more cumbersome for models with additional CIE estimates. The multiple group approach to estimation may present similar challenges (i.e., difficulty interpreting estimates if the dynamic change paths are freely varied across groups with many groups and many time points), but they also allow further freeing of constraints that may be appropriate in a given research scenario but that are not presented here (e.g., differences in initial level or additive components).

5.4 Limitations and Future Directions

There are several important limitations to consider regarding our work on the LCSMM models presented here. First, there has been limited methodological research regarding LCS models with mediators and even less research regarding LCSMM models specifically. Additionally, methodological research on LCS models is also extremely limited in its consideration of studies with imperfect data or methods (i.e., missingness, attrition, model misspecification, etc.) that are likely to occur in real world research, and the discussions around these topics

are mainly limited to univariate LCS models. There is also a lack of established effect sizes for comparing CIEs with respect to both intra-study magnitude (i.e., comparing CIEs within a given study) and comparisons of CIEs across studies for mediation models in the LCS framework; it is unknown whether established effect sizes for mediation in the regression framework (Miočević, O'Rourke, MacKinnon, & Brown, 2018) translate to the LCS framework as well.

In terms of future research directions, an important future direction will involve consideration of moderators with various measurements in LCSMM models: Continuous moderators, time-varying moderators, and moderators of mediation paths when X, M, and Y are all longitudinal and therefore each has univariate dual change structures. The same consideration of calculation of CIEs should be given to these different specifications of LCSMM models. Furthermore, future methodological research should give consideration to how model misspecification impacts statistical power and bias in estimates of CIEs when the theory of change for the b path is misspecified (constrained vs. freely estimated) and when the assumptions of mediation are not met.

5.5 Conclusion

There are many model specification choices to be made that influence the estimation and interpretation of results from models fit within the LCS framework. Some choices are universal to multivariate LCS models, and other choices are specific to inclusion of moderators and mediators into LCS models. Some of the general choices that influence model estimation and interpretation are: measurement of variables (all variables measured longitudinally, or one cross-sectional predictor predicting a longitudinal outcome); specification of longitudinal bivariate relations (coupling, prior level predicting later level, or prior change predicting later change); and both univariate and bivariate dynamic change parameter constraints (proportional change and/or coupling specified to be equal across time, or parameters freely estimated across time). The estimation and subsequent calculation and interpretation of CIEs in LCSMM models varies depending on each of the individual choices made during model specification. Often many of these choices are pre-determined for us with respect to the structure of our data, but several of the choices depend on either the theories about change underlying a research question or the technical propriety of a particular option.

The present paper provides examples that demonstrate obtaining these CIEs in LCSMM models under two different conditions for coupling from M to Y. We also highlight some important considerations related to different components of estimation of CIEs when both moderators and mediators are present in a LCS model. LCSMM models provide researchers with the opportunity to examine more refined theories of mechanisms of change over time, particularly when that change is dynamic and when there are group differences in mechanisms of dynamic change.

References

Altman, D. G., & Royston, P. (2006). The cost of dichotomising continuous variables. *BMJ*, *332*(7549), 1080.1. doi: https://doi.org/10.1136/bmj.332.7549.1080

Alwin, D. F., & Hauser, R. M. (1975). The decomposition of effects in path analysis. *American Sociological Review*, *40*(1), 37. doi: https://doi.org/10.2307/2094445

Baltagi, B. H. (2021). *Econometric Analysis of Panel Data*. Cham: Springer International Publishing. doi: https://doi.org/10.1007/978-3-030-53953-5

Baron, R. M., & Kenny, D. A. (1986). The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of Personality and Social Psychology*, *51*(6), 1173–1182. doi: https://doi.org/10.1037/0022-3514.51.6.1173

Buil, J. M., Van Lier, P. A. C., Brendgen, M. R., Koot, H. M., & Vitaro, F. (2017). Developmental pathways linking childhood temperament with antisocial behavior and substance use in adolescence: Explanatory mechanisms in the peer environment. *Journal of Personality and Social Psychology*, *112*(6), 948–966. doi: https://doi.org/10.1037/pspp0000132

Cheung, G. W., Cooper-Thomas, H. D., Lau, R. S., & Wang, L. C. (2021). Testing moderation in business and psychological studies with latent moderated structural equations. *Journal of Business and Psychology*, *36*(6), 1009–1033. doi: https://doi.org/10.1007/s10869-020-09717-0

Cronbach, L. J., & Furby, L. (1970). How we should measure "change": Or should we? *Psychological Bulletin*, *74*(1), 68–80. doi: https://doi.org/10.1037/h0029382

Cáncer, P. F., & Estrada, E. (2023). Effectiveness of the deterministic and stochastic bivariate latent change score models for longitudinal research. *Structural Equation Modeling: A Multidisciplinary Journal*, *30*(4), 618–632. doi: https://doi.org/10.1080/10705511.2022.2161906

Cáncer, P. F., Estrada, E., & Ferrer, E. (2023). A dynamic approach to control for cohort differences in maturation speed using accelerated longitudinal designs. *Structural Equation Modeling: A Multidisciplinary Journal*, *30*, 761–777. doi: https://doi.org/10.1080/10705511.2022.2163647

Cáncer, P. F., Estrada, E., Ollero, M. J. F., & Ferrer, E. (2021). Dynamical properties and conceptual interpretation of latent change score models. *Frontiers in Psychology*, *12*, 696419. doi: https://doi.org/10.3389/fpsyg.2021.696419

Estrada, E., Bunge, S. A., & Ferrer, E. (2023). Controlling for cohort effects in accelerated longitudinal designs using continuous- and discrete-time dynamic models. *Psychological Methods*, *28*(2), 359–378. doi: https://doi.org/10.1037/met0000427

Goldsmith, K. A., MacKinnon, D. P., Chalder, T., White, P. D., Sharpe, M., & Pickles, A. (2018). Tutorial: The practical application of longitudinal structural equation mediation models in clinical trials. *Psychological Methods*, *23*(2), 191–207. doi: https://doi.org/10.1037/met0000154

Gonzalez, O., & Valente, M. J. (2023). Accommodating a latent XM interaction in statistical mediation analysis. *Multivariate Behavioral Research*, *58*(4), 659–674. doi: https://doi.org/10.1080/00273171.2022.2119928

Gradinger, P., Yanagida, T., Strohmeier, D., & Spiel, C. (2015). Prevention of cyberbullying and cyber victimization: Evaluation of the ViSC social competence program. *Journal of School Violence*, *14*(1), 87–110. doi: https://doi.org/10.1080/15388220.2014.963231

Griffiths, S., Kievit, R. A., & Norbury, C. (2022). Mutualistic coupling of vocabulary and non-verbal reasoning in children with and without language disorder. *Developmental Science*, *25*(3), e13208. doi: https://doi.org/10.1111/desc.13208

Grimm, K. J. (2007). Multivariate longitudinal methods for studying developmental relationships between depression and academic achievement. *International Journal of Behavioral Development*, *31*(4), 328–339. doi: https://doi.org/10.1177/0165025407077754

Grimm, K. J., An, Y., McArdle, J. J., Zonderman, A. B., & Resnick, S. M. (2012). Recent changes leading to subsequent changes: Extensions of multivariate latent difference score models. *Structural Equation Modeling: A Multidisciplinary Journal*, *19*(2), 268–292. doi: https://doi.org/10.1080/10705511.2012.659627

Hamagami, F., & McArdle, J. J. (2001). Advanced studies of individual differences linear dynamic models for longitudinal data analysis. In G. A. Marcoulides & R. E. Schumacker (Eds.), *New Developments and Techniques in Structural Equation Modeling* (pp. 223–266). Psychology Press. doi: https://doi.org/10.4324/9781410601858-13

Hayes, A. F. (2015). An index and test of linear moderated mediation. *Multivariate Behavioral Research*, *50*(1), 1–22. doi: https://doi.org/10.1080/00273171.2014.962683

Hayes, A. F. (2018). Partial, conditional, and moderated moderated mediation: Quantification, inference, and interpretation. *Communication Monographs*, *85*(1), 4–40. doi: https://doi.org/10.1080/03637751.2017.1352100

Hayes, A. F. (2022). *Introduction to mediation, moderation, and conditional process analysis: A regression-based approach* (3rd ed.). New York ; London: The Guilford Press.

Hilley, C. D., & O'Rourke, H. P. (2022). Dynamic change meets mechanisms of change: Examining mediators in the latent change score framework. *International Journal of Behavioral Development*, *46*(2), 125–141. doi: https://doi.org/10.1177/01650254211064352

Hsiao, C. (2014). *Analysis of panel data* (3rd ed.). New York, NY: Cambridge University Press.

Judd, C. M., & Kenny, D. A. (1981). Process analysis: Estimating mediation in treatment evaluations. *Evaluation Review*, *5*(5), 602–619. doi: https://doi.org/10.1177/0193841X8100500502

Jöreskog, K. G. (1971). Simultaneous factor analysis in several populations. *Psychometrika*, *36*(4), 409–426. doi: https://doi.org/10.1007/BF02291366

Könen, T., & Karbach, J. (2021). Individual differences in intervention-related changes. *Advances in Methods and Practices in Psychological Science*, *4*(1), 251524592097917. doi: https://doi.org/10.1177/2515245920979172

MacKinnon, D. P. (2008). *Introduction to statistical mediation analysis*. Routledge.

MacKinnon, D. P., Fritz, M. S., Williams, J., & Lockwood, C. M. (2007). Distribution of the product confidence limits for the indirect effect: Program PRODCLIN. *Behavior Research Methods*, *39*(3), 384–389. doi: https://doi.org/10.3758/BF03193007

MacKinnon, D. P., Lockwood, C. M., Hoffman, J. M., West, S. G., & Sheets, V. (2002). A comparison of methods to test mediation and other intervening variable effects. *Psychological Methods*, *7*(1), 83–104. doi: https://doi.org/10.1037/1082-989X.7.1.83

MacKinnon, D. P., Lockwood, C. M., & Williams, J. (2004). Confidence limits for the indirect effect: Distribution of the product and resampling methods. *Multivariate Behavioral Research*, *39*(1), 99–128. doi: https://doi.org/10.1207/s15327906mbr3901_4

MacKinnon, D. P., & Pirlott, A. G. (2015). Statistical approaches for enhancing causal interpretation of the M to Y relation in mediation analysis. *Personality and Social Psychology Review*, *19*(1), 30–43. doi: https://doi.org/10.1177/1088868314542878

MacKinnon, D. P., Valente, M. J., & Gonzalez, O. (2020). The correspondence between causal and traditional mediation analysis: the link is the mediator by treatment interaction. *Prevention Science*, *21*(2), 147–157. doi: https://doi.org/10.1007/s11121-019-01076-4

McArdle, J. J. (2001). A latent difference score approach to longitudinal dynamic structural analysis. In *Structural Equation Modeling: Present and Future. A Festschrift in Honor of Karl Joreskog* (pp. 341–380). Lincolnwood, IL: Scientific Software International.

McArdle, J. J. (2009). Latent variable modeling of differences and changes with longitudinal data. *Annual Review of Psychology*, *60*(1), 577–605. doi: https://doi.org/10.1146/annurev.psych.60.110707.163612

McArdle, J. J., & Grimm, K. J. (2010). Five steps in latent curve and latent change score modeling with longitudinal data. In K. Van Montfort, J. H. Oud, & A. Satorra (Eds.), *Longitudinal Research with Latent Variables* (pp. 245–273). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: https://doi.org/10.1007/978-3-642-11760-2_8

McArdle, J. J., & Hamagami, F. (1996). Multilevel models from a multiple group structural equation perspective. In *Advanced structural equation modeling: Issues and techniques* (pp. 89–124). Psychology Press.

McArdle, J. J., & Prindle, J. J. (2008). A latent change score analysis of a randomized clinical trial in reasoning training. *Psychology and Aging*, *23*(4), 702–719. doi: https://doi.org/10.1037/a0014349

Miočević, M., O'Rourke, H. P., MacKinnon, D. P., & Brown, H. C. (2018). Statistical properties of four effect-size measures for mediation models. *Behavior*

Research Methods, *50*(1), 285–301. doi: https://doi.org/10.3758/s13428-017-0870-1

Muthén, L. K., & Muthén, B. O. (2017). *Mplus.* Los Angeles, CA: Muthén & Muthén.

O'Rourke, H. P., Fine, K. L., Grimm, K. J., & MacKinnon, D. P. (2022). The importance of time metric precision when implementing bivariate latent change score models. *Multivariate Behavioral Research*, *57*(4), 561–580. doi: https://doi.org/10.1080/00273171.2021.1874261

O'Rourke, H. P., & MacKinnon, D. P. (2015). When the test of mediation is more powerful than the test of the total effect. *Behavior Research Methods*, *47*(2), 424–442. doi: https://doi.org/10.3758/s13428-014-0481-z

O'Rourke, H. P., & MacKinnon, D. P. (2018). Reasons for testing mediation in the absence of an intervention effect: A research imperative in prevention and intervention research. *Journal of Studies on Alcohol and Drugs*, *79*(2), 171–181. doi: https://doi.org/10.15288/jsad.2018.79.171

Preacher, K. J., Rucker, D. D., & Hayes, A. F. (2007). Addressing moderated mediation hypotheses: Theory, methods, and prescriptions. *Multivariate Behavioral Research*, *42*(1), 185–227. doi: https://doi.org/10.1080/00273170701341316

R Core Team. (2020). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria.

Raykov, T. (1999). Are simple change scores obsolete? an approach to studying correlates and predictors of change. *Applied Psychological Measurement*, *23*(2), 120–126. doi: https://doi.org/10.1177/01466219922031248

Rosseel, Y. (2012). lavaan : An R package for structural equation modeling. *Journal of Statistical Software*, *48*(2). doi: https://doi.org/10.18637/jss.v048.i02

Sardeshmukh, S. R., & Vandenberg, R. J. (2017). Integrating moderation and mediation: A structural equation modeling approach. *Organizational Research Methods*, *20*(4), 721–745. doi: https://doi.org/10.1177/1094428115621609

Selig, J. P., & Preacher, K. J. (2009). Mediation models for longitudinal data in developmental research. *Research in Human Development*, *6*(2-3), 144–164. doi: https://doi.org/10.1080/15427600902911247

Serang, S., Grimm, K. J., & Zhang, Z. (2019). On the correspondence between the latent growth curve and latent change score models. *Structural Equation Modeling: A Multidisciplinary Journal*, *26*(4), 623–635. doi: https://doi.org/10.1080/10705511.2018.1533835

Simone, M., & Lockhart, G. (2019). Empirical sample size guidelines for use of latent difference score mediation. *Structural Equation Modeling: A Multidisciplinary Journal*, *26*(4), 636–645. doi: https://doi.org/10.1080/10705511.2018.1540934

Sobel, M. E. (1982). Asymptotic confidence intervals for indirect effects in structural equation models. *Sociological Methodology*, *13*, 290. doi: https://doi.org/10.2307/270723

Usami, S., Hayes, T., & McArdle, J. J. (2016). Inferring longitudinal relationships between variables: Model selection between the latent change score and autoregressive cross-lagged factor models. *Structural Equation Modeling: A Multidisciplinary Journal*, *23*(3), 331–342. doi: https://doi.org/10.1080/10705511.2015.1066680

Valeri, L., & VanderWeele, T. J. (2013). Mediation analysis allowing for exposure–mediator interactions and causal interpretation: Theoretical assumptions and implementation with SAS and SPSS macros. *Psychological Methods*, *18*(2), 137–150. doi: https://doi.org/10.1037/a0031034

Wald, A. (1943). Tests of statistical hypotheses concerning several parameters when the number of observations is large. *Transactions of the American Mathematical Society*, *54*(3), 426–482. doi: https://doi.org/10.1090/S0002-9947-1943-0012401-3

Zaccoletti, S., Camacho, A., Correia, N., Aguiar, C., Mason, L., Alves, R. A., & Daniel, J. R. (2020). Parents' perceptions of student academic motivation during the COVID-19 lockdown: A cross-country comparison. *Frontiers in Psychology*, *11*, 592670. doi: https://doi.org/10.3389/fpsyg.2020.592670

Zhu, S., Sagherian, K., Wang, Y., Nahm, E.-S., & Friedmann, E. (2021). Longitudinal moderated mediation analysis in parallel process latent growth curve modeling in intervention studies. *Nursing Research*, *70*(3), 184–192. doi: https://doi.org/10.1097/NNR.0000000000000503

Journal of Behavioral Data Science, 2025, *5* (2), 28–42.
DOI: https://doi.org/10.35566/jbds/wyman51

Evaluating the Threat of Phantom Faces in Emotion Detection AI through Simulation

Austin Wyman[1] and Zhiyong Zhang[1]

[1]Department of Psychology, University of Notre Dame, Notre Dame, USA
awyman@nd.edu

Abstract. Emotion detection AI is an emerging tool in the field of psychology that enables researchers to process large batches of images of human faces and obtain estimates of the emotions present within images. Some algorithms, such as Py-Feat, are even capable of detecting multiple faces within an image and providing differential estimates for each face. However, a known problem with multiple detection algorithms is that they sometimes mistakenly detect multiple faces when only a single face exists. In such cases, detection of the true face is still available to users and the false face can be ignored, but there may be artifacts of the false face within the true face that are biasing the estimation of emotions. The present study investigated whether the presence of a second face reduces the accuracy of emotion estimation in the first face. Using 1,438 images from the RAVDESS labeled emotion data set, we generated image with multiple faces under a variety of conditions (i.e., size, opacity, emotion similarity, and number of faces) and compared them against unaltered, single face versions of the images. There were meaningful differences in accuracy across between the single-face and multiple-face images, with similarity and number of faces being the most detrimental conditions for multiple-face accuracy. Findings suggest that it is highly important for researchers to remove extraneous faces within images in order to maximize the accuracy of emotion detection analysis.

Keywords: Emotion Detection · Emotion Recognition · Artificial Intelligence · Distortion · Phantom Faces · Multiple Faces

1 Introduction

1.1 Introduction to emotion detection AI

Emotion detection AI (also known as emotion recognition API) is an emerging application of artificial intelligence (AI) used to detect, label, and understand human emotions from images and videos. The technology has been applied in a variety of clinical and educational research settings (Wyman & Zhang, 2023).

Within clinical research, emotion detection AI is often used to develop automated interventions, which monitor participants' emotions and respond with stimuli to influence behavior (Alharbi & Huang, 2020; Bharatharaj, Huang, Mohan, Al-Jumaily, & Krägeloh, 2017; Grossard et al., 2017; Jiang et al., 2019; Liu, Wu, Zhao, & Luo, 2017; Manfredonia et al., 2018). Within educational research, the technology has been used to monitor emotions in response to educational interventions, such as online learning (Chu, Tsai, Liao, & Chen, 2017; Chu, Tsai, Liao, Chen, & Chen, 2020), which are concurrently taking place. However, emotion detection AI is not limited to these applications. In fact, several disciplines in the social and behavioral sciences could benefit from its implementation. Emotion detection AI itself is the integration of research across multiple disciplines, including psychology, physiology, and computer science (Wyman & Zhang, 2025). The technology is based on the concept of action units (AUs, Ekman & Friesen, 1976), which are the simplest combinations of muscles required to produce a facial expression. For example, AU4 corresponds to the act of lowering one's brow and requires the depressor glabellae, depressor supercilli, and corrugator supercilli—three muscles located in the forehead. The Facial Action Coding System (FACS, Ekman & Friesen, 1978) assigns basic emotions to the combination of AUs. For example, when AU4 "brow lowerer" is combined with "upper lid raiser" (AU5), "lid tightener" (AU7), and "lip tightener" (AU23), the facial expression for anger is produced. The FACS traditionally included six emotions—happiness, sadness, anger, surprise, disgust, and fear—but future models were extended to include more emotions like contempt and confusion.

Modern emotion detection AI operationalizes the FACS through a two-step convolutional neural network (CNN), in which the first step of the network focuses on face recognition and the second step on emotion classification. CNNs are a class of artificial neural networks that specialize in processing grid-like topology (Baduge et al., 2022), such as image data, which are treated as a two-dimensional grid of pixels. CNNs are often used to identify patterns within image, such as to detects edges in shapes (Dorafshan, Thomas, & Maguire, 2018), transcribe text from images (Wei, Sheikh, & Ab Rahman, 2018), or recognize faces (Lawrence, Giles, Tsoi, & Back, 1997). CNNs are uniquely suited for processing image data because of their aptitude for handling sparsity. Neural networks are powerful prediction models because they are able to handle multiple layers of parameters that explain complex, often non-linear relationships in the data. However, estimating thousands to millions of parameters when only tens to hundreds are meaningful is computationally expensive (Goodfellow, Bengio, Courville, & Bengio, 2016). The problem is particularly defined for image data, as traditional neural networks are inefficient to the handle the sparse data caused by background and non-focal pixels. CNNs address this problem through a convolution step, which obtains summaries of pixels given by their surrounding information and prioritizes pools of pixels with the most information. In Py-Feat and similar emotion detection AI models (Wyman & Zhang, 2025), the purpose of CNNs is to identify the location of facial features in an image, which is fed to a secondary neural network to determine if action units are activated. Finally, a probabilis-

tic model is conducted, which estimates the probability that a given emotion is being observed given the activated action units. Some emotion detection AI models provide a discrete classification based on the emotion with the highest probability estimate, assuming that an image can only depict one emotion at a time. Other models assume that humans exhibit multiple emotions at once, providing the raw probability estimates for each emotion. Although, different emotion detection AI models use the output in different ways, they all adhere to the same CNN architecture.

1.2 Introduction to Py-Feat

Another difference between emotion detection AI models is whether the model is open-source or commercially-based, which impacts the amount of pre-trained data that is available and the degree of user customizability (Wyman & Zhang, 2025). The Python Facial Expression Analysis Toolbox (Py-Feat, Cheong et al., 2023) is emerging as a valuable open-source model for emotion detection AI, which was created by psychologists for psychologists. The toolbox features 7 emotions (happiness, sadness, anger, surprise, disgust, fear, and neutral) and each emotion is rated continuously on a 0-1 decimal scale. It allows for multiple emotions per image, with each emotion rating representing the proportion of the total face that is exhibiting the given emotion. The Py-Feat architecture consists of five building blocks, which represent different steps of the facial expression analysis procedure. Each block is controlled by a pre-trained, open-source model and can be exchanged by the user for a different model. By default, Py-Feat provides one pre-trained model for face and facial pose estimation, three for facial landmark detection, two for action unit detection, two for emotion detection, and one for identity detection. In particular, the default emotion detection model is the Residual Masking Network (ReMaskNet, Pham, Vu, & Tran, 2021), which Cheong et al. (2023) demonstrated performs better on images in the wild than some commercial emotion detection AI models like iMotions.

Most emotion detection AI models are evaluated using posed images, or images that whose lighting, positioning, and background are carefully designed as to not disrupt the algorithm. However, posed images are not realistic representations of how emotion detection AI models are used in the field, which is why most models have reduced accuracy for images in the wild. Py-Feat has been validated under benchmarked datasets of images in the wild and also has completed robustness tests against common image barriers to facial expression analysis, such as luminance, occlusion, and head rotation (Cheong et al., 2023). Luminance describes the impact of various lighting conditions, either extreme brightness or darkness, which may inhibit the detection of facial landmarks and AUs. Cheong et al. (2023) found that Py-Feat was robust against issues related to luminance on both ends of the spectrum. Occlusion describes the partial obstruction of facial features by an object blocking the face, which similarly inhibits the detection of facial landmarks and AUs. Py-Feat encounters a substantial decline in performance in response to face occlusion (Cheong et al., 2023). Accuracy for face detection, AU detection, and emotion detection models all declined if either

the eyes, nose, or mouth of an image were hidden. Finally, head rotation refers to the direction in the which face is facing the camera in an image. Models are often trained with faces that directly face the camera, but images in the wild are rarely facing straight forward. Models often have challenges detecting the facial landmarks and AUs of side-facing images, which may not generalize to training data. Py-Feat demonstrated robustness against the issue of head rotation (Cheong et al., 2023). The toolbox is a valuable resource for facial expression analysis, as it has been trained and validated on images in the wild, which are more representative of actual usage.

1.3 Phantom faces in emotion detection AI

Aside from traditional image distortions (i.e., luminance, occlusion, and rotation), there is a rare image distortion that has been observed by users of Py-Feat but has not been formally documented in the literature. The distortion is related to the face and facial pose estimation component of Py-Feat and it incorrectly identifies a secondary face within the true face of an image, often located on the forehead of the true face. Figure 1 presents an example of the issue with an image from a benchmarked dataset (Livingstone & Russo, 2018). Note that Py-Feat produces confidence estimates for each face that it detects. The primary face is detected with a confidence of 99.9%, whereas the second, false face is detected with a confidence of 79.9%. The high confidence to detect a second face is concerning given the lack of a second face altogether. No literature exists to define the issue of two faces, nor does it offer any explanation. Hence, given its apparitional appearance, we refer to the issue as "phantom faces".

An easy solution to phantom faces is adjusting the threshold for face detection. For example, the phantom face in the example image had a confidence of 79.9%. By setting the confidence threshold to 0.8 or higher, no analysis would be conducted for any faces below the confidence threshold. However, this solution ignores the issue rather than solving it, as artifacts of the phantom face may remain within the true face even after filtering it out. It is difficult to remove the influence of the phantom face from an image without knowledge of what causes the issue. Therefore, the priority of research should be to identify the extent to which phantom faces bias estimation of emotions in the true faces. Moreover, the issue of biased estimation extends to other cases of emotion detection with multiple faces. Given the frequency of multiple faces in real-world images, often as figures in the background of landscapes or experiments, it is important to understand the extent to which non-focal faces bias the primary face. An empirical examination of the impact of phantom faces and multiple faces on emotion estimation would greatly improve the experimental considerations and practices regarding emotion detection AI and improve the quality of research published in the field.

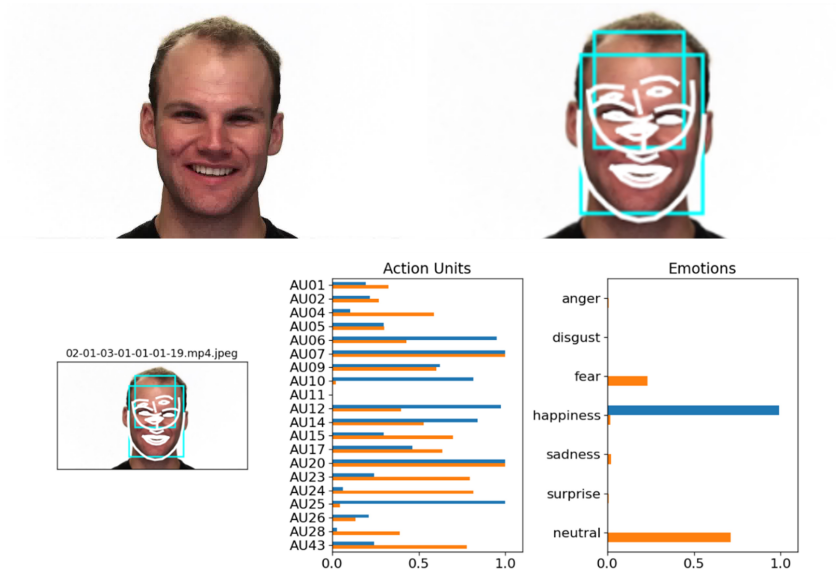

Figure 1. Example of "phantom face" distortion appearing within an image and its corresponding emotion detection AI output. Note. (top left) Original image (02-01-03-01-01-01-19) sampled from a video in the RAVDESS dataset. (top right) Image after processing by Py-Feat, identifying two faces. (bottom) Complete Py-Feat output with blue bars indicating the first identified face (true face) and orange indicating the second face (phantom face). The true face is correctly identified as happy whereas the phantom face is identified as neutral, with low probability of fear, happiness, and sadness.

1.4 Model evaluation for emotion detection AI

Given that the issue of phantom faces has not been discussed in the literature, there is no existing framework for evaluating emotion detection AI models with respect to phantom faces. Currently, models are evaluated using labeled image datasets, which specify a correct response that emotion detection models should be able to match. For example, the dataset may include an image labeled "happy" and for the model is get the case correct it must also produce a "happy" label. Models are evaluated by their accuracy, or how many labels they can correctly match, and their accuracy under various conditions. The conditions are often artificially induced by editing the labeled image. For example, Cheong et al. (2023) created artificial occlusion in images by editing a black bar to cover either the eyes, ears, or mouth of the subject in the image, and manipulated the brightness of the image with a filter to simulate luminance conditions. Yang et al. (2021) similarly applied a Gaussian Blur to images to simulate motion blur and noise from cameras. Some studies also evaluate emotion detection AI models using benchmarked datasets that are designed to include images containing distortions (Kuruvayil & Palaniswamy, 2022; Mollahosseini, Hasani, & Mahoor,

2017). The approach is simple and can be easily replicated across multiple studies. However, using a benchmarked dataset is only accessible when there exists a large collection of images with the intended distortion. When images do not exist, it is necessary to design an experiment and recruit participants, which can be expensive. Moreover, the process of labeling new image data can be onerous based on the large sample size necessary to make stable inferences regarding model performance. Simulating distortions in images is more accessible to answer certain questions related to emotion detection AI model evaluation.

1.5 Present study

Currently, there is no existing benchmarked dataset that describes phantom faces or any issues related to multiple face detection, meaning the question of their impact on emotion detection AI models must be evaluated through a simulated data. The purpose of the present study is to understand the risk of phantom faces or multiple faces in classification tasks using emotion detection AI. Distortions like facial occlusion reduce the accuracy of emotion detection AI models by blocking the estimation of AUs. Phantom faces may also block necessary facial landmarks and interfere with AUs. Therefore, the primary hypothesis is that the presence of phantom faces leads to a decrease in accuracy in emotion classification. To study this, the present study develops a novel experiment for simulating phantom faces, which may be replicated by other researchers evaluating emotion detection AI models. Emotion detection AI may be a significant technology for advancing the emotion research in the social and behavioral sciences, but its success is dependent on the support of rigorous frameworks for model evaluation.

2 Methods

2.1 Materials and procedures

The present study utilized image data from the publicly available Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS, Livingstone & Russo, 2018) dataset. RAVDESS contains 7,356 video and audio recordings. The dataset contains both speech and song recordings, but the present study only utilizes the subset of speech recordings. There are 4,320 speech recordings available, which are divided into full audio-video, video only, and audio only recordings. Only the 1,440 video only recordings were relevant to emotion detection. During the experiment, actors were instructed to vocalize two statements ("Kids are talking by the door", "Dogs are sitting by the door") with eight emotional intentions (neutral, calm, happy, sad, angry, fearful, surprise, and disgust) and two emotional intensities (normal, strong); however, conditions involving the neutral emotion were only vocalized with normal intensity. Actors repeated each vocalization twice, resulting in 60 total trials per actor ($N = 24$).

Actors were recruited from the Toronto, Canada area with ages ranging from 21 to 33. Odd-numbered actors were male ($n = 12$) and even-numbered actors

were female ($n = 12$). Additionally, actors represented Caucasian ($n = 20$), East-Asian ($n = 2$), and multiracial ($n = 2$) ethnic backgrounds. Actors did not have any distinctive facial features, such as facial hair or tattoos, and were instructed not to wear glasses, in order to ensure minimal interference with face detection algorithms. There are multiple benchmarking datasets available for emotion detection AI, but the RAVDESS dataset was selected as an appropriate dataset because of its highly structured system of labeled data, which allow researchers to evaluate the accuracy of models at various conditions (e.g., statement, intensity).

The RAVDESS dataset only includes videos of participants. Images were extracted as frames from each trial using the "av" package in R (Ooms, 2024a), sampling at a frame rate of three frames per second. The sampling produced an average of 11 images per video with a standard deviation of one image. Although all images in a video share the same label, the images may not be equally representative of the label. For example, a video may contain the emotion anger but the actor may not be presenting anger the entire time, as there may be resting emotions also captured on camera before or after. Therefore, it is important to isolate the most representative frame of the target emotion (i.e., the emotion labeled by the dataset). Py-Feat emotion estimates were obtained for each image within a trial. The image with the maximum estimate for the target emotion was selected as the most representative frame of the video and used for the present analysis. The final sample contained 1,246 images, which consisted of the most representative frames from the 1,440 videos excluding the trials labeled as "calm", given that Py-Feat is not able to detect calm emotions.

2.2 Simulation design

The present study created control and experiment sets of images based on the RAVDESS sample. The set of control images were unedited from the original RAVDESS images, depicting a singular face. The set of experiment images were identical to the control set except that they were edited to include an additional face. Within the experiment set, the control image was the primary focus of the camera and an additional face was selected from the RAVDESS sample to appear somewhere in the background close to the face, but without occlusion of the face. Phantom faces were inserted into the experiment set using the "magick" package in R (Ooms, 2024b) and were edited according to a variety of conditions: size, opacity, number, and sameness. The size condition describes the size of the phantom face in the image, with the phantom face appearing as either 25% or 50% of the size of the focal image. The opacity condition describes the lack of transparency of the phantom face, with the phantom face appearing fully visible at 0% opacity, half visible at 50% opacity, and near completely transparent at 75% opacity. The number condition describes the number of phantom faces that appears in the image. The number condition 1 inserts a first phantom face into the top left corner of the image, 2 inserts a second phantom face in the top right, 3 inserts the third face in the bottom left, and 4 inserts the fourth in the bottom right. A location condition is somewhat nested within the number condition, but

the location of phantom face should not matter as long as the phantom face is equally obstructing the primary face in all four conditions. Phantom faces are carefully positioned as to avoid occlusion; therefore, location is not a meaningful condition. Finally, the sameness condition determines whether the phantom face exhibits the same emotion as the primary face and, if the emotions are different, which emotion is exhibited. The values of the sameness condition are "same", "anger", "disgust", "fear", "happiness", "neutral", "sadness", and "surprise." In the "same" condition, phantom faces inserted were identical to the primary face. In the remaining emotion conditions, phantom faces that were most representative of the target emotion were inserted. The most representative images were again identified from RAVDESS and were selected by which image the maximum target emotion estimates in the sample. The present study explored 192 total simulation conditions.

2.3 Data analysis

The present study examined the difference in performance of Py-Feat on images in the control set and experiment set. Performance was primarily measured by the overall classification accuracy of each image set, parameter bias in emotion estimation, and the conditional classification accuracy within each simulation condition. Differences in overall accuracy were evaluated using a paired t-test. Given the similarity of images in the control and experiment set, a paired t-test is appropriate because the difference in accuracy is approximately normal at large samples and sample variances are approximately equal. A similar t-test approach was used to evaluate parameter bias in emotion estimates for each emotion label across the 192 conditions. Finally, differences in conditional accuracy were evaluated using an Analysis of Variance (ANOVA) model to investigate the main effects of each variable and their two-way interaction effects. All analyses were conducted in R version 4.4.1.

3 Results

Py-Feat correctly identified labeled emotions in 81.3 percent of control set images. Since the control set was not subjected to any simulation conditions, the control set accuracy was consistent across all conditions. Py-Feat demonstrated substantially lower accuracy in the experiment set of images. The average accuracy for the experiment set was 54.7 percent with a standard deviation of 18.5 percent. The average difference in accuracy between the control and experiment set was 26.6 percent, which the paired t-test demonstrated was statistically significant, $t(191) = 19.9, p < .001$. Moreover, Cohen's d was 1.44, indicating that the presence of phantom faces in images substantially impacts Py-Feat's overall classification accuracy.

Py-Feat's classification accuracy is also impacted by the condition of phantom face images. The main effect of size on the difference in accuracy between control and experiment sets of images was statistically significant, $F(1, 190) =$

89.8, $p < .001$. Py-Feat performed worse at labeling emotions with phantom faces that were 50% of the size of the primary face (Mean difference = 0.37, SD = 0.18) than phantom faces that were 25% (M = 0.16, SD = 0.12). The main effect of sameness on accuracy difference was also significant, $F(7, 184) = 37.5, p < .001$. When phantom images exhibited the same emotion as the primary face, there was a small difference in accuracy between the control and experiment set (M = 0.05, SD = 0.06). However, there were more pronounced differences when the primary and phantom faces exhibited different images. The largest difference was observed when the phantom face exhibited happiness (M = 0.49, SD = 0.15), followed by anger (M = 0.42, SD = 0.15), fear (M = 0.37, SD = 0.13), disgust (M = 0.25, SD = 0.09), neutral (M = 0.23, SD = 0.04), surprise (M = 0.18, SD = 0.16), and sadness (M = 0.13, SD = 0.12). The interaction effect of size and sameness was also statistically significant, $F(7, 176) = 41.6, p < .001$, indicating that the magnitude of bias caused by the size of phantom images varied depending on the emotion of the phantom face. Table 1 presents a summary of the Tukey Honest Significance Difference test comparisons, which examined the simple effects of the interaction. The difference between the 25% and 50% conditions was greatest within the happiness and surprise conditions, but the 50% condition performed significantly worse across the different levels of the sameness condition. In contrast, the main effects of opacity, $F(2, 189) = 0.5, p = 0.542$, and number, $F(3, 188) = 0.1, p = 0.985$, were not statistically significant and nor were their interaction effects with any of the other variables. Table 2 presents the highest and lowest differences in accuracy conditions, corroborating the claim that phantom face accuracy is largely determined by size and sameness.

Table 1. Tukey Honest Significance Difference test contrasts among size conditions within sameness conditions.

Emotion	Mean difference (SE)	t statistic
Same	-0.076 (0.014)	-5.25 ***
Anger	-0.274 (0.014)	-18.79 ***
Disgust	-0.172 (0.014)	-11.81 ***
Fear	-0.256 (0.014)	-17.58 ***
Happiness	-0.303 (0.014)	-20.78 ***
Neutral	-0.071 (0.014)	-4.89 ***
Sad	-0.22 (0.014)	-15.06 ***
Surprised	-0.30 (0.014)	-20.71 ***

Note. *** $p < .001$. All t statistics were obtained with 176 degrees of freedom.

Differences in continuous emotion estimates between images in the control and experiment set were also examined to examine the effect of phantom faces on parameter estimation. Each emotion was examined separately. All means are expressed in units of Py-Feat estimates, which range from 0-1. The largest difference was observed among anger estimates, $d = -0.15$. There was a -0.04 mean difference in anger estimates, $t(276095) = -77.6, p < .001$, indicating that Py-

Table 2. Best case and worst case scenarios for phantom face conditions.

Case Number	Difference	Size	Opacity	Number	Emotion
1	0.006	25%	0%	1	Same
2	0.007	25%	0%	2	Same
3	0.007	25%	0%	4	Same
190	0.651	50%	50%	3	Happiness
191	0.651	50%	50%	2	Happiness
192	0.652	50%	0%	3	Happiness

Feat overestimates the anger of images when phantom faces are present. A 0.04 mean difference in neutral estimates was observed, $t(276095) = 43.7, p < .001$. The effect size was positive, $d = 0.14$, indicating that Py-Feat underestimates neutral emotions in the presence of phantom faces. There was a -0.03 mean difference in happiness estimates, $t(276095) = -47.2, p < .001$, with an effect size of $d = -0.09$. There was a 0.02 mean difference in surprise estimates, $t(276095) = 43.7, p < .001$, with an effect size of $d = 0.08$. There was a 0.01 mean difference in sadness estimates, $t(276095) = 33.3, p < .001$, with an effect size of $d = 0.06$. There was a -0.01 mean difference in fear estimates, $t(276095) = -23.9, p < .001$, with an effect size of $d = -0.04$. Finally, there was a 0.008 mean difference in disgust estimates, $t(276095) = 14.8, p < .001$, with an effect size of $d = 0.03$. All differences were statistically significant, but were classified as small effect sizes.

4 Discussion

4.1 Findings and implications

Py-Feat demonstrated lower accuracy at classifying images in the experiment set than the control set, and the difference was statistically significant. However, it is a known property of statistical tests that as sample size becomes increasingly large, statistical tests will always converge toward a statistically significant result, regardless of how menial the practical significance of the result is (Meehl, 1967). Therefore, we should prioritize the effect size of results because it is unaffected by sample size. The effect size of the first paired t-test was 1.44, which is substantially larger than Cohen's threshold for a large effect size (0.8). Therefore, we are confident that the observed difference in accuracy between the control and experiment set is practically significant as well. The difference in overall classification accuracy is sufficient to claim that phantom faces are a valid threat to the inference of emotion detection AI. Py-Feat users that conduct analysis on images that contain phantom faces can expect a substantial reduction in accuracy.

However, the effect of bias was not uniformly distributed across the simulation conditions. Some phantom faces conditions, such as 25% size and same emotion, resulted in less than a 1 percent difference in accuracy between the

control set and experiment set. Other conditions, such as 50% size and happiness, resulted in a massive 65 percent difference in accuracy. Opacity had no main or interaction effects with accuracy difference, suggesting that Py-Feat's algorithm is sophisticated enough to detect facial landmarks and AUs regardless of how transparent the image is. It seems that as long as Py-Feat is able to detect the phantom face at all, the phantom face biases the image. Similarly, neither the number main effect nor its interactions were significant, suggesting that there is no multiplicative impact of multiple phantom faces. The presence of one phantom face alone is enough to bias the image. The lack of a significant effect associated with number also suggests that there is no significant effect of phantom face location, as long as the phantom face is not directly obstructing the primary face. However, it is important to note that there were no safe conditions observed. The presence of a phantom face in the experiment set always produced lower accuracy than their control set baseline, yet various conditions determined the severity of the bias that was observed.

Py-Feat consistently performed better in the presence of smaller phantom faces than larger phantom faces, but the impact was differentially observed for different clusters of emotions. The smallest difference was observed for same and neutral emotions, which is an intuitive result. In the same condition, the phantom face was a duplicate of the primary face, except smaller, and therefore, it was only capable of biasing the primary face with its own AUs. The only bias that could be produced by the same condition is that which is caused by occlusion, which was intentionally limited in the experiment design. The neutral emotion is unique from other motions in the FACS because it is defined by the lack of any AUs activated at all. Therefore, AUs from a neutral phantom face were not able to interfere with the AUs of the primary face, as the phantom face AUs did not exist. The larger accuracy difference due to happiness can also be explained by AUs. Happiness is one of the simplest facial expressions to explain by AUs, consisting of only two AUs: "cheek raiser" (AU6) and "lip corner puller" (AU12). Additionally, the two AUs are not repeated in any other emotion, making the presence of the two AUs with any combination of other AUs an easy decision to label the image as happiness. Consequently, emotion detection AI models tend to have the highest accuracy classifying happiness labels, with some models even achieving 100 percent accuracy in benchmarked datasets (Yang et al., 2021). Therefore, it is likely that Py-Feat defaults to classifying happiness emotions, which it has the highest accuracy for, when it detects the necessary AUs in the phantom face, even if they do not appear in the primary face. The next large decrease in accuracy was caused by the surprise condition, which is an emotion that shares multiple AUs with fear and anger—two other emotions that resulted in a large decrease. It is likely that Py-Feat confused the three emotions because of their similarity, as it combined AUs from both the primary face and phantom face. The difference between the three emotions may have been more pronounced at larger phantom face sizes than smaller because the AUs were available in a higher resolution, making them easier to detect.

Paired t-test results for the continuous estimates were all statistically significant, but it is not recommended to rely on statistical significance given the large number of simulation cases ($N = 276095$). The effect size estimates paint a different story, finding that the true differences in emotion estimates were all approximately 0. The small effect is likely due to the aggregation across all simulation cases, including cases in which neither primary face nor the phantom face exhibited the emotion of interest. However, it is important to note that the difference in continuous estimates was not unidirectional. Anger, happiness, and fear were overestimated by Py-Feat in the presence of phantom faces, whereas neutral, surprise, sadness, and disgust were underestimated. The results corroborate the previous claims that happiness estimates are detected more often because of their AUs and that anger and fear are often mistaken for surprise in phantom faces. However, the effect of phantom faces conditions on continuous estimates remains unknown, which may influence the severity of bias.

4.2 Limitations and future directions

The present study contributed a novel experiment design framework for evaluating the issue of phantom faces in emotion detection AI; however, there are multiple limitations to the current design. Phantom faces cannot be directly replicated in an image because what causes phantom faces to appear naturally is unknown. Therefore, it is uncertain whether the experiment set of images is representative of phantom faces encountered in the wild, but it certainly is generalizable to the broader problem of multiple faces in emotion detection. Multiple faces appear in images under a variety of circumstances, whether they are a passing figure in the background or an active backdrop of an experiment (e.g., classrooms). The present study identified the risk of including any non-primary face in the background of emotion detection tasks, which results in a substantial decrease in classification accuracy and an increase in bias for continuous emotion estimation. Researchers conducting emotion detection AI work should prioritize removing the influence of any non-primary faces from the image before conducting any analysis.

The present study introduced the issue of phantom faces and examined its risks, but it did not provide any empirical solutions for addressing the problem. Researchers could crop images around the primary face to remove the influence of other faces. However, cropping images would not address phantom faces that appear within the primary face. Additionally, cropping images may not be feasible when important information is contained within the background of an image. Future research should investigate other methods for removing the influence of other faces, such as background blurring or targeted face blurring, which may not have such tradeoffs.

Another limitation of the present simulation design is that the experiment set of images was only evaluated under four variables, some of which containing only 2 or 3 levels. Future research may want to investigate other size and opacity parameters than the ones selected in the present simulation. Additionally, there may be other factors that influence the severity of phantom faces, which

were not considered in the present study. Future research should identify these factors and expand the simulation paradigm of emotion detection AI to evaluate its robustness across a diversity of conditions. Finally, this study only evaluated the performance of Py-Feat but several other emotion detection AI models are available, such as Amazon Rekognition and Google Cloud AI, and can be investigated in the future. Future research should observe how the problem of phantom faces replicates across other models and the novel solutions that may emerge.

4.3 Conclusion

Emotion detection AI is an emerging technology in the social and behavioral sciences, which may transform the accessibility of multimodal designs in emotion research; however, the current technology is limited by the lack of rigorous methodology for AI model evaluation. Simulation studies using edited images revealed insight into the problem of phantom faces and multiple faces, but they may provide insight into other challenges with emotion detection AI models as well. The paradigm of simulation studies has bolstered quantitative methodology by elucidating the circumstances in which methods flounder or flourish. It can be applied just as eagerly to AI model evaluation, provided that at ground truth is known, such as with labeled data sets. The present simulation introduced the presence of phantom faces as a substantive issue, which we hope motivates other researchers to identify possible solutions. Through the continuation of rigorous evaluation work, emotion detection AI may become a valuable tool for emotion research.

Acknowledgments

The study was presented at the 2025 Annual Meeting of the International Society for Data Science and Analytics. Wyman is supported by the NSF Graduate Research Fellowship (2236418), the Notre Dame Program for Interdisciplinary Education Research Burns Fellowship, and the Lucy Graduate Scholars Program. Zhang is supported by the US Department of Education (R305D210023) and Notre Dame Global. The authors certify that they have no conflicts of interests to declare that are relevant to the content of this article.

References

Alharbi, M., & Huang, S. (2020). An augmentative system with facial and emotion recognition for improving social skills of children with autism spectrum disorders. In *2020 IEEE International Systems Conference (SysCon)* (pp. 1–6). doi: https://doi.org/10.1109/SysCon47679.2020.9275659

Baduge, S. K., Thilakarathna, S., Perera, J. S., Arashpour, M., Sharafi, P., Teodosio, B., ... Mendis, P. (2022). Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications. *Automation in Construction*, *141*, 104440. doi: https://doi.org/10.1016/j.autcon.2022.104440

Bharatharaj, J., Huang, L., Mohan, R. E., Al-Jumaily, A., & Krägeloh, C. (2017). Robot-assisted therapy for learning and social interaction of children with autism spectrum disorder. *Robotics*, *6*(1), 4. doi: https://doi.org/10.3390/robotics6010004

Cheong, J. H., Jolly, E., Xie, T., Byrne, S., Kenney, M., & J, C. L. (2023). Py-Feat: Python facial expression analysis toolbox. *Affective Science*, *4*, 781–796. doi: https://doi.org/10.1007/s42761-023-00191-4

Chu, H.-C., Tsai, W.-H., Liao, M.-J., & Chen, Y.-M. (2017). Facial emotion recognition with transition detection for students with high-functioning autism in adaptive e-learning. *Soft Computing*, *22*, 2973–2999. doi: https://doi.org/10.1007/s00500-017-2549-z

Chu, H.-C., Tsai, W.-H., Liao, M.-J., Chen, Y.-M., & Chen, J.-Y. (2020). Supporting e-learning with emotion regulation for students with autism spectrum disorder. *Educational Technology & Society*, *23*(4), 124–146. Retrieved from https://www.jstor.org/stable/26981748

Dorafshan, S., Thomas, R. J., & Maguire, M. (2018). Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, *186*, 1031–1045. doi: https://doi.org/10.1016/j.conbuildmat.2018.08.011

Ekman, P., & Friesen, W. V. (1976). Measuring facial movement. Environmental psychology & nonverbal behavior. *Journal of Personality and Social Psychology*, *1*(1), 56–75. doi: https://doi.org/10.1007/BF01115465

Ekman, P., & Friesen, W. V. (1978). *Facial action coding system.* American Psychological Association (APA). doi: https://doi.org/10.1037/t27734-000

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning.* Cambridge: MIT Press.

Grossard, C., Grynspan, O., Serret, S., Jouen, A.-L., Bailly, K., & Cohen, D. (2017). Serious games to teach social interactions and emotions to individuals with autism spectrum disorders (ASD). *Computers & Education*, *113*, 195–211. doi: https://doi.org/10.1016/j.compedu.2017.05.002

Jiang, M., Francis, S. M., Srishyla, D., Conelea, C., Zhao, Q., & Jacob, S. (2019). Classifying individuals with asd through facial emotion recognition and eye-tracking. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 6063–6068). doi: https://doi.org/10.1109/EMBC.2019.8857005

Kuruvayil, S., & Palaniswamy, S. (2022). Emotion recognition from facial images with simultaneous occlusion, pose and illumination variations using meta-learning. *Journal of King Saud University-Computer and Information Sciences*, *34*(9), 7271–7282. doi: https://doi.org/10.1016/j.jksuci.2021.06.012

Lawrence, S., Giles, C., Tsoi, A. C., & Back, A. (1997). Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks*, *8*(1), 98–113. doi: https://doi.org/10.1109/72.554195

Liu, X., Wu, Q. J., Zhao, W., & Luo, X. (2017). Technology-facilitated diagnosis and treatment of individuals with autism spectrum disor-

der: An engineering perspective. *Applied Sciences*, *7*(10), 1051. doi: https://doi.org/10.3390/app7101051

Livingstone, S. R., & Russo, F. A. (2018). The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, *13*(5), e0196391. doi: https://doi.org/10.1371/journal.pone.0196391

Manfredonia, J., Bangerter, A., Manyakov, N. V., Ness, S., Lewin, D., Skalkin, A., ... others (2018). Automatic recognition of posed facial expression of emotion in individuals with autism spectrum disorder. *Journal of Autism and Developmental Disorders*, *49*, 279–293. doi: https://doi.org/10.1007/s10803-018-3757-9

Meehl, P. E. (1967). Theory-testing in psychology and physics: A methodological paradox. *Philosophy of science*, *34*(2), 103–115. doi: https://doi.org/10.1086/288135

Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2017). Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, *10*(1), 18–31. doi: https://doi.org/10.1109/TAFFC.2017.2740923

Ooms, J. (2024a). av: Working with audio and video in r [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=av` (R package version 0.9.3)

Ooms, J. (2024b). magick: Advanced graphics and image-processing in r [Computer software manual]. Retrieved from `https://CRAN.R-project.org/package=magick` (R package version 2.8.5)

Pham, L., Vu, T. H., & Tran, T. A. (2021). Facial expression recognition using residual masking network. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 4513–4519). doi: https://doi.org/10.1109/ICPR48806.2021.9411919

Wei, T. C., Sheikh, U., & Ab Rahman, A. A.-H. (2018). Improved optical character recognition with deep neural network. In *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)* (pp. 245–249). doi: https://doi.org/10.1109/CSPA.2018.8368720

Wyman, A., & Zhang, Z. (2023). API face value: Evaluating the current status and potential of emotion detection software in emotional deficit interventions. *Journal of Behavioral Data Science*, *3*(1), 59–69. doi: https://doi.org/10.35566/jbds/v3n1/wyman

Wyman, A., & Zhang, Z. (2025). A tutorial on the use of artificial intelligence tools for facial emotion recognition in R. *Multivariate Behavioral Research*, *60*(3), 641–655. doi: https://doi.org/10.1080/00273171.2025.2455497

Yang, K., Wang, C., Sarsenbayeva, Z., Tag, B., Dingler, T., Wadley, G., & Goncalves, J. (2021). Benchmarking commercial emotion detection systems using realistic distortions of facial image datasets. *The Visual Computer*, *37*(6), 1447–1466. doi: https://doi.org/10.1007/s00371-020-01881-x

Journal of Behavioral Data Science, 2025, *5* (2), 43–63.
DOI: https://doi.org/10.35566/jbds/xu2025

More Than a Model: The Compounding Impact of Behavioral Ambiguity and Task Complexity on Hate Speech Detection

Shuo Xu[1,†], Hailiang Wang[2,†], Yijun Gao[3], Yixiang Li[4], and Meng-Ju Kuo[5,*]

[1] Computer Science and Engineering Department, University of California San Diego, La Jolla, CA, USA
shx009@ucsd.edu
[2] School of Computer Science, College of Computing, Georgia Institute of Technology, Atlanta, GA, USA
nealgatech@gmail.com
[3] Krieger School of Arts and Sciences, Johns Hopkins University, Washington, DC, USA
gaoyijun818@gmail.com
[4] Department of Computer Science, The George Washington University, Washington, DC, USA
yixiang607@gmail.com
[5] Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA
mengjuk@alumni.cmu.edu

Abstract. The automated detection of hate speech is a critical but difficult task due to its subjective, behavior-driven nature, which leads to frequent annotator disagreement. While advanced models (e.g., transformers) are state-of-the-art, it is unclear how their performance is affected by the methodological choice of label aggregation (e.g., majority vote vs. unanimous agreement) and task complexity. We conduct a 2x2 quasi-experimental study to measure the compounding impact of these two factors: Labeling Strategy (low-ambiguity "Pure" data vs. high-ambiguity "Majority" data) and Task Granularity (Binary vs. Multi-class). We evaluate five models (Logistic Regression, Random Forest, Light Gradient Boosting Machine [LightGBM], Gated Recurrent Unit [GRU], and A Lite BERT [ALBERT]) across four quadrants derived from the HateXplain dataset. We find that (1) ALBERT is the top-performing model in all conditions, achieving its peak F1-Score (0.8165) on the "Pure" multi-class task. (2) Label ambiguity is strongly associated with performance loss; ALBERT's F1-Score drops by ≈15.6% (from 0.8165 to 0.6894) when trained on the higher-disagreement "Majority" data in the multi-class setting. (3) This negative effect is compounded by task complexity, with the performance drop being nearly twice as severe for the multi-class task as for the binary task. A sensitivity analysis confirmed this drop is not an artifact of sample size. We conclude that

in HateXplain, behavioral label ambiguity is a more significant bottleneck to model performance than model architecture, providing strong evidence for a data-centric approach.

Keywords: Hate Speech Detection · Behavioral Data Science · Label Ambiguity · Transformer Models · Text Classification

1 Introduction

The automated detection of hate speech is a critical challenge for online platform governance and social science research (Mansur, Omar, & Tiun, 2023). While deep learning models, particularly transformers, have become the state-of-the-art approach (Malik, Qiao, Pang, & van den Hengel, 2025), their performance is fundamentally dependent on the quality of the human-annotated data used for training. Unlike objective classification tasks like spam detection (Xu et al., 2025), identifying hate speech is a subjective, behavior-driven task. Hate speech is nuanced, context-dependent, and culturally specific, leading to significant and unavoidable disagreements among human annotators (Mathew et al., 2021).

This 'behavioral ambiguity' is a core problem in many applied data science fields, including the detection of fake news (Alghamdi, Lin, & Luo, 2023; Shah & Patel, 2025; Tanvir, Mahir, Akhter, & Huq, 2019; Tian, Xu, Cao, Wang, & Wei, 2025) and the monitoring of mental illness on social media (Cao et al., 2025; Ding et al., 2025; Zhang et al., 2025). A common method for resolving this ambiguity is to aggregate labels using a 'majority vote'. However, this approach can be problematic, as it masks underlying disagreement and treats highly contested labels with the same certainty as those with unanimous agreement. This introduces significant label noise, which can degrade model performance.

Existing studies on hate-speech detection span both classical feature-based pipelines and modern neural architectures. For example, Davidson, Warmsley, Macy, and Weber (2017) study a three-class taxonomy (hate/offensive/neither) and report strong overall performance (overall precision = 0.91, recall = 0.90, F1 = 0.90), while showing that the hate class remains substantially harder (hate-class precision = 0.44; recall = 0.61). On the HateXplain benchmark (the same corpus used in this study), Mathew et al. (2021) report transformer-based baselines, including BERT with macro F1 = 0.674 and area under the receiver operating characteristic curve (AUROC) = 0.843, and a rationale-supervised variant (BERT-HateXplain) with macro F1 = 0.687 and AUROC = 0.851. Because our goal is not to introduce a new architecture but to quantify how label aggregation and task granularity change achievable performance under a fixed pipeline, we focus on within-corpus comparisons across our four experimental conditions rather than direct head-to-head benchmarking against heterogeneous prior pipelines.

Although it is well documented that label noise can reduce supervised classification performance and that multi-class text classification is generally more difficult than binary classification, these general observations are not yet sufficient for behavioral hate-speech annotation settings. In such settings, annotator

disagreement often reflects systematic ambiguity in human judgment rather than purely random error, and aggregation choices (e.g., unanimous filtering vs. majority vote) implicitly change the data-generating process. Moreover, prior studies rarely isolate whether task granularity interacts with disagreement-driven ambiguity to yield a compounding effect under a controlled design that holds the modeling and evaluation pipeline constant.

The central novelty is the explicit estimation of a compounding (interaction) effect, rather than only reporting that ambiguity and complexity each reduce performance in isolation. In particular, we contribute a design-based evaluation that treats label aggregation strategy as a primary methodological factor and estimates both main effects and an ambiguity×granularity interaction. Specifically, we use a 2×2 quasi-experimental framework crossing (i) labeling strategy (unanimous 'Pure' vs. majority-vote 'Majority') and (ii) task granularity (binary vs. multi-class), while keeping the model suite and evaluation protocol fixed across conditions. This positioning shifts the contribution from the general claim that 'label noise hurts' to an explicit characterization of when and how strongly behavioral ambiguity becomes a bottleneck as task definitions become more fine-grained.

Guided by this objective, we structure our inquiry around three research questions. First, we establish a baseline performance (RQ1) by evaluating classical and deep learning models on 'Pure' (unanimous agreement) data. We then measure how model performance changes (RQ2) when the models are trained instead on 'Majority' (ambiguous, high-noise) data. Finally, we explore whether task complexity (Binary vs. Multi-class) compounds the negative effects of label ambiguity (RQ3).

To answer these questions, we evaluate five models (Logistic Regression, Random Forest, Light Gradient Boosting Machine [LightGBM], Gated Recurrent Unit [GRU], and A Lite BERT [ALBERT]) across four distinct datasets derived from the HateXplain corpus (Mathew et al., 2021). Our results demonstrate that while the ALBERT transformer is the top-performing model in all conditions, its performance is (1) highest on 'Pure' data, (2) significantly degraded by 'Majority' data ambiguity, and (3) further compounded by the combination of ambiguity and task complexity. We conclude that data quality, rooted in behavioral agreement, is a more significant bottleneck than model architecture for this task.

2 Methods

To systematically investigate our research questions (RQs) on the impact of label ambiguity and task complexity, we designed and executed a 2×2 quasi-experimental study. The design is quasi-experimental in the sense that factor levels are induced by curating subsets of an existing annotated corpus rather than by randomly assigning instances to conditions. This framework allowed us to isolate and measure the effects of these two key variables on the performance of a diverse range of machine learning (ML) and deep learning (DL) models.

All experiments were conducted in a Python 3 environment, primarily using Google Colab with NVIDIA T4 GPUs. The implementation relied on `pandas` for data management, `scikit-learn` and `lightgbm` for classical ML models, `PyTorch` for the recurrent neural network, and the `Transformers` library for the ALBERT model (G. Lan, Inan, et al., 2025; G. Lan, Zhang, et al., 2025).

2.1 Experimental Design

Our methodology is built around the three Research Questions (RQs) introduced in Section 1 that investigate the interplay between model choice, label ambiguity, and task granularity. To test our hypotheses, we structured our study around two primary factors, creating four distinct experimental quadrants.

The first factor is the Labeling Strategy (Behavioral Dimension), which directly tests the impact of annotator disagreement (i.e., behavioral ambiguity). We defined two levels for this factor. The Pure condition represents a low-noise, high-agreement scenario using only data with unanimous annotator agreement. Conversely, the Majority condition represents a higher-noise, higher-ambiguity scenario using a simple majority vote, which mixes clear and contested labels. We use the term label ambiguity to emphasize that disagreements may reflect systematic differences in human judgment rather than random mistakes; in supervised learning, this manifests as label noise when a single aggregated hard label is used for training despite disagreement. We note that this labeling-strategy factor is induced by deterministic filtering on annotator agreement rather than randomized assignment. Deterministic filtering selects examples based on an explicit rule (here, agreement level) rather than by random sampling; as a result, the resulting subsets can differ systematically in text difficulty and other properties, introducing distribution shift between conditions. As a result, the 'Pure' and 'Majority' subsets may differ not only in target ambiguity (label uncertainty), but also in the distribution of inputs (covariate shift). In particular, unanimous cases are plausibly enriched for more prototypical or straightforward examples, whereas majority-vote cases may include more borderline, context-dependent, or otherwise difficult instances that elicit disagreement. Therefore, the Pure–Majority contrast should be interpreted as a quasi-experimental operationalization of behavioral ambiguity that may also correlate with inherent text difficulty.

The second factor is Task Granularity (Task Dimension), which tests whether task complexity compounds the effects of label noise. This factor also has two levels: Binary Classification, which is a simpler task (aggregating to 'Normal' vs. 'Toxic'), and Multi-class Classification, which is the original, more complex task ('Normal' vs. 'Offensive' vs. 'Hatespeech'). Multi-class classification is typically more difficult than binary classification because the model must learn finer-grained decision boundaries among multiple competing labels. In hate-speech settings, categories such as offensive and hatespeech can be semantically close and context-dependent, which increases both model confusion and annotator disagreement.

This 2×2 design yields four distinct experimental conditions: (1) Binary-Pure, (2) Binary-Majority, (3) MultiClass-Pure, and (4) MultiClass-Majority. By training and evaluating an identical suite of models within each quadrant, we can isolate the performance effects attributable to our two main factors and their interaction. We summarize this experimental framework in Table 1.

Table 1. Summary of the 2×2 Experimental Design.

	Pure (Unanimous)	**Majority (High Ambiguity)**
Binary Task (Normal vs. Toxic)	`Binary-Pure` (Low Noise, Simple Task)	`Binary-Majority` (High Noise, Simple Task)
Multi-class Task (Normal vs. Offensive vs. Hatespeech)	`MultiClass-Pure` (Low Noise, Complex Task)	`MultiClass-Majority` (High Noise, Complex Task)

2.2 Data Source and Curation

We selected the HateXplain dataset (Mathew et al., 2021) as the source corpus for our experiments. The HateXplain dataset used in this study is publicly accessible at Hugging Face. While the original paper focused on explainability (rationales), this corpus is well-suited to our purpose because it provides the individual, non-aggregated annotations for its 20,148 posts. The original authors reported 'moderate agreement' among annotators; our methodology is explicitly designed to treat this observation as a variable to be tested rather than simply a dataset limitation.

The HateXplain release is distributed in JSON format, where each record contains tokenized post content (`post_tokens`) and up to three individual annotator entries (HateXplain typically has three annotators per post). To support reproducibility, we converted the JSON records into a flat tabular structure prior to dataset construction. Specifically, for each record we (i) retained a unique identifier (`id`); (ii) reconstructed a text string by detokenizing `post_tokens` (joining tokens with whitespace and then normalizing spacing around punctuation and brackets); and (iii) extracted per-annotator fields into separate columns, including each annotator's numeric label, mapped string label, annotator identifier, target span(s) when available, and rationales. In HateXplain, the numeric labels were mapped as `0`→`hatespeech`, `1`→`normal`, and `2`→`offensive`. The resulting flattened table therefore contains (a) a reconstructed text field used as model input and (b) up to three separate annotator-label columns (`label1`-`label3`) used to define unanimous-agreement (Pure) versus majority-vote (Majority) subsets.

To support transparency and reproducibility, the preprocessing and dataset-construction scripts (including JSON flattening and deterministic filtering rules) will be released in a public GitHub repository upon acceptance.

We curated our four experimental datasets by operationalizing the Labeling Strategy factor separately for each task type. Because this curation is agreement-based, it may induce systematic differences in example difficulty between the resulting subsets, which we treat as a limitation in interpreting Pure–Majority performance differences. For the Multi-class Tasks, the MultiClass-Pure dataset includes only posts where all three annotators agreed on the specific class (e.g., all three voted 'Normal'). In contrast, the MultiClass-Majority dataset includes posts where at least two of the three annotators agreed on a specific class (a simple majority agreement).

For the **Binary Tasks**, we first consolidated the 'Offensive' and 'Hatespeech' labels into a single 'Toxic' class. The Binary-Pure dataset was then created by requiring unanimous agreement on this binary split (i.e., all three agreed on 'Normal', or all three agreed on one of the 'Toxic' categories). For the Binary-Majority dataset, we first consolidated the labels to binary at the annotator level, resulting in three binary labels per post. We then applied a majority-vote rule on these binary labels, retaining all posts where at least two of the three annotators agreed on the binary outcome.

The resulting class distributions and total sample sizes for each of our four experimental datasets are detailed in Table 2. As shown, all conditions exhibit substantial class imbalance, which informed our choice of evaluation metrics.

Table 2. Total Sample Size and Class Distribution for Each Experimental Dataset.

Experiment	Label Strategy Logic	Normal	Offensive	Hatespeech	Total
Binary-Majority	Majority	7,814	Toxic: 12,334		20,148
Binary-Pure	Pure	5,124	Toxic: 8,637		13,761
MultiClass-Majority	Majority	7,814	5,480	5,935	19,229
MultiClass-Pure	Pure	5,124	1,761	2,960	9,845

2.3 Reader Guide to Key ML/NLP Concepts

In supervised text classification, models learn to predict categorical labels from text using examples annotated by humans. Because models require numeric inputs, text is converted to numeric representations either via sparse feature vectors (e.g., TF–IDF) or via learned dense representations (embeddings) in neural networks. In this study, annotator disagreement is treated as label ambiguity; in machine learning terms, ambiguity can induce label noise because the effective training label depends on the aggregation rule. Finally, because our Pure versus Majority datasets are constructed by deterministic filtering on agreement, the resulting subsets may differ in their input-text distributions (distribution shift) in addition to differing in supervision ambiguity.

2.4 Data Preprocessing and Feature Engineering

A standardized preprocessing pipeline was applied to the raw text of all four datasets to ensure consistency. This pipeline, implemented as a single cleaning function, executed a sequence of four transformations: Lowercasing was performed on all text. Token Replacement was applied using regular expressions to identify social media-specific entities, which were then replaced with special tokens to preserve context (e.g., URLs became `<URL>`, user mentions became `<USER>`, and hashtags became `<HASHTAG>`). Following this, all remaining Special Character Removal of non-alphanumeric and non-whitespace characters occurred. Finally, Whitespace Normalization collapsed multiple whitespace characters into a single space, and leading or trailing whitespace was stripped.

Following this cleaning pipeline, we employed two distinct feature engineering strategies tailored to the different model architectures. For the Classical ML Models (LR, RF, LGBM), the cleaned text was further processed by removing English stopwords (via NLTK) and applying lemmatization. The resulting text was then vectorized using Term Frequency–Inverse Document Frequency (TF–IDF), capturing both individual words (unigrams) and two-word pairs (bigrams) with an `n_gram_range` of (1, 2). In our implementation, TF–IDF features were generated using `scikit-learn`'s `TfidfVectorizer` with `ngram_range=(1,2)` (unigrams and bigrams), `stop_words='english'`, and `max_features`=10,000, resulting in a sparse feature vector of up to 10,000 dimensions per document (depending on the fitted vocabulary).

TF–IDF (term frequency–inverse document frequency) represents each document as a numeric vector where each dimension corresponds to a word or short phrase and the value reflects how important that term is in the document relative to the corpus. We include unigrams (single words) and bigrams (two-word phrases) because short phrases can capture meaning that is not recoverable from individual tokens alone. To avoid information leakage, the TF–IDF vectorizer vocabulary and inverse-document-frequency weights were fit on the training split only; the fitted vectorizer was then applied to transform the validation and test splits.

For the Deep Learning Models (GRU, ALBERT), we used the cleaned text directly (without stopword removal or lemmatization) to preserve the full sequential context. For the GRU, a custom vocabulary was built, and sequences were tokenized and padded to a fixed length. For the GRU pipeline, text was tokenized at the word level via whitespace splitting after normalization (lowercasing; replacing URLs/users/hashtags with special markers; removing non-alphanumeric characters; whitespace normalization). A vocabulary of size `VOCAB_SIZE`=10,000 was built from the training split only by selecting the 9,999 most frequent tokens and mapping them to indices 1–9,999; index 0 was reserved for both padding and out-of-vocabulary (OOV) tokens. Sequences were truncated to `max_length`=200 tokens and padded with 0s using `torch.nn.utils.rnn.pad_sequence` (batch_first).

For ALBERT, the text was fed directly into the pre-trained `Albert-base-v2` tokenizer, with sequences truncated or padded consistent with the pre-trained checkpoint's requirements. In both deep learning cases, the models learned their

own latent representations directly from these token sequences during training. For ALBERT, we used the `AlbertTokenizer` from the `albert-base-v2` checkpoint with `max_len`=200, applying `truncation=True` and `padding='max_length'` to produce fixed-length inputs. Fine-tuning was performed with mini-batches of size 32. No layers were frozen (i.e., all model parameters were updated during training).

2.5 Model Architectures

To test our research questions, we selected a suite of five models representing a wide spectrum of learning strategies, from interpretable linear models to complex non-linear ensembles and state-of-the-art contextual deep learning models.

2.5.1 Machine Learning Models We selected three classical ML models to serve as strong baselines. These models are widely used in text classification and are well-suited for high-dimensional, sparse TF–IDF feature matrices. They represent three distinct approaches to classification: linear models, bagging ensembles, and boosting ensembles.

Logistic Regression (LR) As a robust and highly interpretable linear baseline, we used Logistic Regression (Hosmer & Lemeshow, 2000). LR models the probability of a discrete outcome by fitting a linear combination of the input features (the TF–IDF vectors) to a logistic (sigmoid) function, generalized using the softmax function for the multi-class task. We used the `scikit-learn` implementation with ℓ_2 (Ridge) regularization to prevent overfitting and manage multicollinearity.

Tree-Based Ensemble Models We implemented two powerful non-linear ensemble models, grouped by their core ensemble strategy: bagging and boosting. Our bagging model of choice was the Random Forest (RF) (Breiman, 2001), which constructs a large number of decorrelated decision trees in parallel through the use of bootstrapped samples and random feature subsets. The final prediction is determined by a majority vote across trees, which effectively reduces variance. Our boosting model of choice was LightGBM (LGBM) (Ke et al., 2017), a highly efficient and scalable implementation of gradient boosting (Friedman, 2001). Unlike RF's parallel approach, gradient boosting builds trees sequentially, with each new tree trained to correct the residual errors of the existing ensemble. LightGBM utilizes a leaf-wise growth strategy and histogram-based splitting, making it particularly efficient on large, sparse feature spaces.

2.5.2 Deep Learning Models We selected two deep learning architectures to assess the performance of models that learn representations directly from raw sequential text, rather than from pre-computed TF–IDF features.

Gated Recurrent Unit (GRU) To represent sequential models, we used a Gated Recurrent Unit (GRU) network (Cho et al., 2014). GRUs are a variant of recurrent neural networks (RNNs) that process text token by token, using update and reset gates to control the flow of information, allowing the model to capture long-range dependencies. Our model, implemented in `PyTorch`, consisted of an embedding layer, a multi-layer GRU encoder, and a final linear layer for classification.

ALBERT (A Lite BERT) To represent the state of the art in contextual modeling, we fine-tuned ALBERT (Z. Lan et al., 2020). ALBERT is an efficient variant of the transformer model BERT that uses parameter sharing and factorized embeddings to reduce model size while maintaining high performance. Unlike the sequential GRU, ALBERT processes the entire text sequence in parallel, building deep, bidirectional contextual representations of each token. We fine-tuned the Albert-base-v2 checkpoint from the Hugging Face Transformers library for our classification tasks (Rao Killi, Balakrishnan, & Rao, 2024). Transformers differ from RNN/GRU models in that they rely on self-attention mechanisms (rather than recurrence) to model relationships between all tokens in a sequence. Self-attention allows each token representation to directly attend to other tokens, enabling efficient parallel computation and effective modeling of long-range dependencies.

For neural models, tokenization converts text into a sequence of discrete units (tokens). Tokens are mapped to numeric vectors (embeddings) that the model learns or fine-tunes during training. Because sequences vary in length, inputs are truncated to a maximum length and padded with a special token to enable minibatch training. Tokens that are not present in the training vocabulary are treated as out-of-vocabulary (OOV) and mapped to a dedicated OOV token.

2.6 Model Training and Hyperparameter Tuning

A central component of our methodology was ensuring a fair, 'apples-to-apples' comparison between the computationally inexpensive classical models and the more resource-intensive deep learning models. To this end, we enforced a consistent data-splitting and model-selection protocol across all four experimental quadrants. For each quadrant, the data were split into 60% training, 20% validation, and 20% test sets. Splits were stratified by class to preserve distributions, and a fixed random seed ensured that all models used the exact same partition. No model used information from the test set during training or hyperparameter tuning.

All reported scores are from the single held-out 20% test set. The training and tuning process was standardized as follows: All models (LR, RF, LGBM, GRU, ALBERT) were initially trained on the 60% `train` set for a grid or random sample of hyperparameters. Hyperparameter configurations were evaluated using performance on the 20% `validation` set, with Weighted F1-Score as the primary selection criterion. The single best configuration per model architecture (i.e., the one achieving the highest validation Weighted F1) was selected.

2.6.1 Hyperparameter Search Strategies We employed different search strategies for the two categories of models to balance thoroughness and computational cost.

For the Machine Learning Models (LR, RF, LGBM), we performed an exhaustive grid search using `GridSearchCV` from `scikit-learn` to identify the optimal combination of hyperparameters. Cross-validation folds were drawn only from the training partition. The specific parameter grids used for the classical models are detailed in Table 3.

Table 3. Hyperparameter Grids for Machine Learning Models (Grid Search).

Model	Hyperparameter	Values Searched
LR	`C`	[0.1, 1, 10]
	`solver`	[‘`lbfgs`’, ‘`saga`’]
	`penalty`	[‘`l2`’]
	`class_weight`	[‘`balanced`’, None]
RF	`n_estimators`	[50, 100, 200]
	`max_depth`	[None, 10, 20]
	`min_samples_split`	[2, 5, 10]
	`min_samples_leaf`	[1, 2, 4]
	`class_weight`	[‘`balanced`’, None]
LGBM	`n_estimators`	[100, 200]
	`learning_rate`	[0.01, 0.1, 0.2]
	`num_leaves`	[31, 63]
	`max_depth`	[None, 10]
	`class_weight`	[‘`balanced`’, None]

For the Deep Learning Models (GRU, ALBERT), an exhaustive grid search was computationally infeasible. We instead employed a random search of 10 iterations to efficiently explore the hyperparameter space. For each sampled configuration, models were trained on the training set and evaluated on the validation set. The ranges from which hyperparameters were drawn are detailed in Table 4. Both DL models were optimized with standard variants of the Adam optimizer and trained with mini-batches; early stopping based on validation performance was used where applicable to prevent overfitting.

2.7 Evaluation Metrics and Confidence Intervals

Given the substantial class imbalance in all quadrants (Table 2), we focused on metrics that account for class support. We prioritize F1-Score over accuracy because accuracy can be inflated by the majority class under substantial imbalance and does not reflect minority-class performance. F1-Score is the harmonic mean of precision and recall, and the weighted version aggregates class-wise F1

Table 4. Hyperparameter Ranges for Deep Learning Models (Random Search).

Model	Hyperparameter	Ranges Sampled
GRU	`embedding_dim`	$[150, 250]$ (Integer)
	`hidden_dim`	$[256, 768]$ (Integer)
	`lr` (learning rate)	$[10^{-4}, 10^{-3}]$ (Log-uniform)
	`epochs`	$[5, 10]$ (Integer)
ALBERT	`lr` (learning rate)	$[10^{-5}, 3 \times 10^{-5}]$ (Log-uniform)
	`epochs`	$[3, 6]$ (Integer)
	`dropout`	$[0.0, 0.25]$ (Uniform)

values using class support so that each class contributes proportionally to its frequency. In contrast, Macro-AUC treats each class equally by averaging one-vs-rest AUCs without weighting, which is informative when minority classes are substantively important. For both binary and multi-class tasks, we report the Weighted Precision, Weighted Recall, and Weighted F1-Score, computed as the support-weighted average of class-specific values. We also report the area under the receiver operating characteristic curve (AUROC or AUC) for binary tasks and Macro-AUC for multi-class tasks, where Macro-AUC is defined as the unweighted mean of one-vs-rest ROC AUCs across classes.

To quantify uncertainty without assuming a specific parametric distribution for the metrics, we computed 95% **confidence intervals (CIs)** for F1 and AUC via a non-parametric bootstrap of the test set, stratified by class. We used a stratified bootstrap to preserve the original class proportions in each resample, which stabilizes uncertainty estimates in the presence of class imbalance and reduces the chance that minority classes are underrepresented in bootstrap samples. For each model and quadrant, we generated $B = 1000$ bootstrap resamples of the test data (with replacement), refit the metric on each resample, and reported the 2.5th and 97.5th percentiles of the resulting empirical distribution as the CI.

All results reported in Section 3 correspond to the held-out 20% test set.

3 Results

This section presents the empirical findings from our 2×2 experimental framework. The results are organized by our research questions, first establishing the baseline performance in low-noise 'Pure' conditions (RQ1), and then analyzing the impact of label ambiguity and task complexity (RQ2 & RQ3). All reported scores are from the held-out 20% test set.

3.1 RQ1: Baseline Performance on 'Pure' Data

Our first research question sought to establish baseline model performance under ideal, low-noise conditions. To answer this, we evaluated all five models on the 'Pure' datasets, where all annotators were in unanimous agreement.

3.1.1 Performance on the Binary-Pure Task In the simpler binary classification task (N=13,761), the deep learning models demonstrated a clear advantage over the classical TF-IDF-based models. Table 5 details the performance of all models. The **ALBERT transformer model** was the clear top performer, achieving a Weighted F1-Score of 0.8126 (95% CI $[0.7980, 0.8270]$). The GRU model also performed strongly with an F1-Score of 0.7877 (95% CI $[0.7732, 0.8046]$), notably outperforming the best classical model, Random Forest (F1: 0.7356). The non-overlapping confidence intervals suggest a clear performance gap between the deep learning models and the classical models.

Table 5. Model Performance on the **Binary-Pure** Test Set. Metrics are weighted averages for Precision and Recall. Best scores are in bold.

	Weighted Metrics			
Model	**Precision**	**Recall**	**F1-Score (95% CI)**	**AUC (95% CI)**
LR	0.71	0.72	0.7138 [0.6975, 0.7309]	0.7791 [0.7622, 0.7963]
RF	0.74	0.74	0.7356 [0.7179, 0.7522]	0.7860 [0.7684, 0.8035]
LGBM	0.73	0.74	0.7323 [0.7159, 0.7485]	0.7929 [0.7754, 0.8093]
GRU	0.79	0.79	0.7877 [0.7732, 0.8046]	0.8619 [0.8481, 0.8751]
ALBERT	0.81	0.82	**0.8126 [0.7980, 0.8270]**	**0.8835 [0.8694, 0.8965]**

3.1.2 Performance on the MultiClass-Pure Task We observed a similar pattern in the more granular 'MultiClass-Pure' task (N=9,845), as detailed in Table 6. ALBERT once again achieved the highest performance across all metrics, with a Weighted F1-Score of 0.8165 (95% CI $[0.7997, 0.8338]$) and a Macro-AUC of 0.9226 (95% CI $[0.9118, 0.9328]$). Interestingly, the GRU model (F1: 0.7367) did not perform as well in this multi-class setting, falling behind the classical ensemble models. The best classical model was Random Forest (F1: 0.7730). The confidence intervals for ALBERT do not overlap with any other model, indicating a clear and substantial performance advantage.

These baseline results from both 'Pure' quadrants confirm our first hypothesis: under ideal, high-agreement data conditions, the deep contextual representations learned by the transformer model (ALBERT) provide a measurable performance advantage over both sequential (GRU) and TF-IDF-based classical models.

Table 6. Model Performance on the **MultiClass-Pure** Test Set. Metrics are weighted averages for Precision and Recall. Best scores are in bold.

	Weighted Metrics			
Model	**Precision**	**Recall**	**F1-Score (95% CI)**	**Macro-AUC (95% CI)**
LR	0.77	0.76	0.7646 [0.7451, 0.7840]	0.8796 [0.8665, 0.8928]
RF	0.77	0.77	0.7730 [0.7549, 0.7921]	0.8875 [0.8746, 0.9001]
LGBM	0.78	0.76	0.7673 [0.7480, 0.7860]	0.8792 [0.8658, 0.8915]
GRU	0.74	0.73	0.7367 [0.7177, 0.7547]	0.8497 [0.8351, 0.8650]
ALBERT	**0.82**	**0.81**	**0.8165 [0.7997, 0.8338]**	**0.9226 [0.9118, 0.9328]**

3.2 RQ2: The Impact of Label Ambiguity on Performance

Our second research question addresses the effect of behavioral ambiguity (label noise) on model performance. To answer this, we evaluated all models on the 'Majority' datasets and compared their performance to the 'Pure' (low-noise) baselines.

3.2.1 Performance on the Binary-Majority Task We first analyzed the Binary-Majority task (N=20,148). As shown in Table 7, the introduction of label ambiguity resulted in a universal and clear performance degradation for every model compared to the 'Pure' condition (Table 5).

ALBERT remained the top-performing model with a Weighted F1-Score of 0.7447 (95% CI $[0.7304, 0.7576]$). However, this represents a substantial $\approx 8.4\%$ relative decrease from its 0.8126 F1-Score on the 'Pure' data. The GRU model (F1: 0.7196) also saw a significant drop from its 'Pure' performance (F1: 0.7877) but remained the clear second-best performer, outperforming all classical models.

Table 7. Model Performance on the **Binary-Majority** Test Set. Metrics are weighted averages for Precision and Recall. Best scores are in bold.

	Weighted Metrics			
Model	**Precision**	**Recall**	**F1-Score (95% CI)**	**AUC (95% CI)**
LR	0.66	0.67	0.6637 [0.6488, 0.6784]	0.7154 [0.7005, 0.7310]
RF	0.69	0.70	0.6887 [0.6746, 0.7035]	0.7279 [0.7124, 0.7425]
LGBM	0.68	0.69	0.6810 [0.6668, 0.6957]	0.7365 [0.7216, 0.7520]
GRU	0.72	0.72	0.7196 [0.7060, 0.7337]	0.7941 [0.7801, 0.8082]
ALBERT	**0.74**	**0.75**	**0.7447 [0.7304, 0.7576]**	**0.8272 [0.8141, 0.8400]**

3.2.2 Performance on the MultiClass-Majority Task A similar, and even more pronounced, drop in performance was observed in the MultiClass-Majority task (N=19,229), as shown in Table 8.

Again, ALBERT remained the best performer, but its F1-Score fell to 0.6894 (95% CI [0.6742, 0.7037]). This is a massive ≈ 15.6% relative decrease from its 0.8165 score in the clean 'MultiClass-Pure' task (Table 6). In this high-noise environment, the classical models (LR, RF, LGBM) were all clustered around ≈ 0.65 F1, while the GRU model (F1: 0.5984) struggled significantly, performing the worst of all models.

Table 8. Model Performance on the **MultiClass-Majority** Test Set. Metrics are weighted averages for Precision and Recall. Best scores are in bold.

	Weighted Metrics			
Model	**Precision**	**Recall**	**F1-Score (95% CI)**	**Macro-AUC (95% CI)**
LR	0.66	0.66	0.6518 [0.6368, 0.6676]	0.8106 [0.7995, 0.8218]
RF	0.66	0.66	0.6552 [0.6400, 0.6700]	0.8194 [0.8090, 0.8305]
LGBM	0.65	0.65	0.6526 [0.6376, 0.6681]	0.8138 [0.8030, 0.8243]
GRU	0.60	0.60	0.5984 [0.5821, 0.6132]	0.7566 [0.7446, 0.7680]
ALBERT	**0.69**	**0.69**	**0.6894 [0.6742, 0.7037]**	**0.8472 [0.8375, 0.8571]**

These two experiments confirm our second hypothesis: introducing label ambiguity by using a 'Majority' vote strategy severely degrades the performance of all models, regardless of task complexity.

3.3 RQ3: Analysis of the Compounding Impact of Task Complexity

Our third research question explored whether task complexity (Binary vs. Multi-class) interacts with and worsens the negative effect of label ambiguity. By comparing the results from the four tables presented in RQ1 and RQ2, we can analyze this critical interaction effect.

This analysis reveals two key patterns based on two separate measurement axes. We first measure the 'cost of ambiguity' (Horizontal Comparison) for both tasks by comparing the 'Pure' results (Tables 5 and 6) against the 'Majority' results (Tables 7 and 8) for our best model, ALBERT. For the Binary task, introducing ambiguity caused the F1-Score to drop from 0.8126 to 0.7447, resulting in a loss of 0.0679. However, for the more complex Multi-class task, introducing ambiguity caused the F1-Score to drop from 0.8165 to 0.6894[6], a significantly larger loss of 0.1271. This difference demonstrates that the performance drop from label noise was almost twice as severe for the more complex multi-class task.

[6] Because Mathew et al. (2021) reports transformer baselines on the same HateXplain corpus, direct architectural benchmarking is possible; however, our contribution is a controlled within-corpus design that estimates how aggregation strategy (Pure vs. Majority) and task granularity (binary vs. multi-class) shift achievable performance under a fixed pipeline.

The second axis measures the 'cost of complexity' (Vertical Comparison) for both data quality levels. In the 'Pure' (low-noise) condition, increasing the task complexity from Binary to Multi-class had a negligible impact on ALBERT's performance (F1-Score was stable at 0.8126 vs. 0.8165). This suggests the ALBERT model is robust to task complexity when the underlying data quality is high. In sharp contrast, in the 'Majority' (high-noise) condition, increasing task complexity caused a significant performance drop, from 0.7447 to 0.6894.

These two analyses confirm our third hypothesis. The negative impact of label ambiguity is compounded by task complexity: while the transformer model is robust to complexity on clean data, it is highly sensitive to complexity when data quality is low. This interaction effect is a key finding of our 2×2 experiment.

3.4 Sensitivity Analysis: Disentangling Data Quality from Sample Size

A primary critique of our findings could be that the performance gap between the 'Pure' and 'Majority' conditions is confounded by sample size. For instance, the MultiClass-Pure dataset (N=9,845) is smaller than the MultiClass-Majority dataset (N=19,229). One could argue that the 'Pure' models performed better simply because the dataset was smaller and easier to model.

To rule out this confound, we conducted a sensitivity analysis. We created a `Majority-Downsampled` dataset by taking a random subsample of the MultiClass-Majority data to match the exact size of the MultiClass-Pure dataset (N=9,845). We then trained and evaluated our key models on this new dataset, which had low quality (high noise) but an identical sample size to the 'Pure' data.

The implication of this analysis was consistent. Model performance on the randomly sampled Majority-Downsampled dataset was nearly identical to the performance observed on the full MultiClass-Majority dataset. For instance, ALBERT achieved an F1-Score of approximately 0.69 on the downsampled set, which is dramatically lower than its 0.8165 score achieved on the MultiClass-Pure set. This analysis provides strong evidence that the observed performance gap is not an artifact of sample size. However, because the Pure and Majority conditions are constructed via agreement-based filtering, this sensitivity analysis does not rule out systematic distribution shift: the Majority condition may contain a higher proportion of intrinsically difficult or context-dependent texts. We therefore interpret the Pure–Majority performance gap as reflecting disagreement-driven ambiguity and the harder-case distribution associated with annotator disagreement, rather than label ambiguity alone.

3.5 Summary of Findings

To summarize the entire 2×2 experiment, Table 9 presents the best-performing model (based on Weighted F1-Score) from each of the four experimental quadrants. The results clearly show a **'performance cliff'**.

While ALBERT was the top-performing model in all four conditions, its performance was highest in the MultiClass-Pure condition (F1: **0.8165**), suggesting that the model benefits from fine-grained, high-quality labels. Performance drops significantly when label ambiguity is introduced ('Majority' data), and drops further still when task complexity is added on top of that ambiguity (the MultiClass-Majority condition), confirming our core hypotheses.

Table 9. Summary of Best Weighted F1-Scores (ALBERT) Across the 2×2 Experimental Design.

		Factor 1	
		'Pure' Data (Low Ambiguity)	**'Majority' Data** (High Ambiguity)
Factor 2	**Binary Task**	0.8126	0.7447
	Multi-class Task	**0.8165**	0.6894

4 Discussion

The results from our 2 × 2 experimental design provide clear answers to our research questions and offer significant insights into the impact of behavioral ambiguity on hate speech detection. Our discussion is organized into three parts: first, we address the critical confound of sample size; second, we interpret the main findings in the context of our hypotheses; and third, we discuss the broader implications and avenues for future work.

4.1 Addressing the Confound of Sample Size (Sensitivity Analysis)

A primary critique of our findings could be that the performance gap between the 'Pure' and 'Majority' conditions is confounded by sample size. For instance, the MultiClass-Pure dataset (N=9, 845) is much smaller than the `MultiClass-Majority` dataset (N=19, 229). One might argue that the 'Pure' models performed better simply because the dataset was smaller and easier to model.

To rule out this confound, we conducted a sensitivity analysis. We created a `Majority-Downsampled` dataset by taking a random subsample of the MultiClass-Majority data to match the exact size of the MultiClass-Pure dataset (N=9, 845). We then trained and evaluated our key models on this new dataset, which had *low quality (high noise)* but an *identical sample size* to the 'Pure' data.

The indications of this sensitivity analysis were clear: model performance on the random sampled `Majority-Downsampled` dataset was nearly identical to the performance observed on the full MultiClass-Majority dataset. For example, ALBERT achieved an F1-Score of approximately 0.69 on the downsampled set,

which is dramatically lower than its 0.8165 score achieved on the MultiClass-Pure set. This analysis provides strong evidence that the performance gap is genuinely attributable to **data quality (i.e., label ambiguity)**, not the artifact of sample size.

4.2 Interpretation of Main Findings

With the sample size confound addressed, we can interpret the main results from our 2×2 experiment.

4.2.1 RQ1: Transformers Excel on 'Pure' Behavioral Data Our results from the 'Pure' conditions (Tables 5 and 6) serve as a clear baseline. The ALBERT model's superior performance ($F_1 > 0.81$) confirms that when the behavioral signal is unambiguous (i.e., unanimous annotator agreement), modern transformers can effectively learn and generalize. The non-overlapping confidence intervals suggest this advantage is not trivial. This finding aligns with the broader consensus that transformers are the state-of-the-art for most NLP tasks.

4.2.2 RQ2: Behavioral Ambiguity and Hard Cases Drive Performance Loss The most significant finding of this study is the dramatic performance drop when moving from 'Pure' to 'Majority' data. In the multi-class task (Table 8), ALBERT's F1-Score plummeted from 0.8165 to 0.6894, a $\approx 15.6\%$ relative decrease. This demonstrates that the 'noise' introduced by simple majority-vote labeling—which mixes clear signals with highly contested ones—is the a major practical bottleneck in this dataset to model performance. This confirms our second hypothesis that label ambiguity severely harms model efficacy. Because the Majority set is constructed via agreement-based filtering, this drop likely reflects both (i) ambiguity in the supervision signal (contested labels) and (ii) a shift toward more borderline or context-dependent texts that are intrinsically harder to classify. Thus, while disagreement is strongly associated with performance loss in our design, we do not claim the effect is attributable to label ambiguity alone.

4.2.3 RQ3: Task Complexity Compounds the Effect of Ambiguity Our third hypothesis was that task complexity would compound the negative effect of noise. The summary in Table 9 clearly supports this by highlighting an interaction effect between the two factors.

When data was 'Pure' (Low Noise), moving from a simple binary task (F_1 : 0.8126) to a complex multi-class task (F_1 : 0.8165) had a negligible impact. This suggests the ALBERT model is robust to task complexity when the data is clean. However, when data was 'Majority' (High Noise), the story changed dramatically. Performance on the simple binary task (F_1 : 0.7447) was already degraded, but when task complexity was added, performance fell even further to 0.6894. This demonstrates a clear interaction: the models are capable of handling

complexity, but not when the training data is also ambiguous and noisy. The negative impact of ambiguity is significantly worsened when combined with high task complexity.

4.3 Implications and Future Work

Our findings have significant implications for the field of behavioral data science and hate speech detection.

The results constitute a powerful argument for the 'Data-Centric' approach to AI. The bottleneck for this behavioral task was clearly not the model architecture (ALBERT was superior) but the quality of the data. Consequently, simply building a bigger model will not solve a problem rooted in ambiguous human-annotated labels.

Furthermore, our results show that the widespread practice of using 'majority vote' is a problematic aggregation strategy. It masks inherent behavioral ambiguity and creates a noisy signal that degrades even the most advanced models. Researchers should be transparent about their aggregation methods and, when possible, use 'Pure' (unanimous) subsets for more reliable model benchmarking.

Finally, this study refines the general understanding of label noise in NLP. While previous literature has broadly established that label noise degrades performance, our findings characterize a specific **interaction effect**: behavioral ambiguity is not merely an additive cost, but a multiplier of difficulty that disproportionately affects fine-grained tasks. This suggests that as the field moves toward more complex, multi-class behavioral taxonomies, the ROI on resolving annotator disagreement will be significantly higher than on architectural optimization.

For future work on ambiguity, researchers should treat annotator disagreement as a *feature*, not as noise to be filtered. Methods from psychology and psychometrics, such as modeling individual annotator biases or using 'think-aloud' protocols, could provide a deeper understanding of why disagreement occurs (Ge, 2024). Furthermore, developing models that can explicitly learn from ambiguous or 'soft' labels could be a fruitful avenue for progress in handling behaviorally complex tasks.

4.4 Scope and Generalizability

Our empirical findings are based on a single corpus (HateXplain) with a specific platform context, annotator pool, language, and annotation protocol. These characteristics can shape both the prevalence of disagreement and the difficulty of the classification task. Accordingly, our results should be interpreted as evidence that, in HateXplain, disagreement-driven ambiguity (and the harder cases associated with it) is strongly linked to performance degradation and that this degradation is amplified in the multi-class setting. Establishing whether the same compounding pattern holds more broadly will require replication across additional hate-speech corpora, platforms, and labeling protocols, including settings with different annotator instructions, community norms, and class definitions.

4.5 Conclusion

This study conducted a 2 × 2 quasi-experimental analysis to measure the impact of behavioral label ambiguity and task complexity on hate speech detection. Our findings demonstrate that while transformer models (ALBERT) are the top performers, their efficacy is fundamentally dependent on data quality. In the HateXplain corpus and its annotation protocol, we find that training under majority-vote aggregation is associated with substantially worse performance, and that this degradation is amplified in the multi-class setting. Given the agreement-based construction, this pattern likely reflects both disagreement-driven ambiguity and the harder cases associated with annotator disagreement. These findings suggest that, for this dataset, improving how disagreement and ambiguity are represented and modeled may be as important as improving model architecture. Further validation on additional corpora and platforms is needed before drawing broader conclusions about hate-speech detection in general.

References

Alghamdi, J., Lin, Y., & Luo, S. (2023). Towards covid-19 fake news detection using transformer-based models. *Knowledge-Based Systems*, *274*, 110642. doi: https://doi.org/10.1016/j.knosys.2023.110642

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. doi: https://doi.org/10.1023/A:1010933404324

Cao, Y., Dai, J., Wang, Z., Zhang, Y., Shen, X., Liu, Y., & Tian, Y. (2025). Machine learning approaches for depression detection on social media: A systematic review of biases and methodological challenges. *Journal of Behavioral Data Science*, *5*(1). doi: https://doi.org/10.35566/jbds/caoyc

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1724–1734). Doha, Qatar: Association for Computational Linguistics. doi: https://doi.org/10.3115/v1/D14-1179

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *Proceedings of the international aaai conference on web and social media (icwsm)* (Vol. 11, pp. 512–515). doi: https://doi.org/10.1609/icwsm.v11i1.14955

Ding, Z., Wang, Z., Zhang, Y., Cao, Y., Liu, Y., Shen, X., ... Dai, J. (2025). Trade-offs between machine learning and deep learning for mental illness detection on social media. *Scientific Reports*, *15*, 14497. doi: https://doi.org/10.1038/s41598-025-99167-6

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232. doi: https://doi.org/10.1214/aos/1013203451

Ge, J. (2024). Technologies in peace and conflict: Unraveling the politics of deployment. *International Journal of Re-*

search Publication and Reviews (IJRPR), *5*(5), 5966–5971. doi: https://doi.org/10.55248/gengpi.5.0524.1273

Hosmer, D. W., & Lemeshow, S. (2000). *Applied logistic regression* (2nd ed.). New York, NY: John Wiley & Sons.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st international conference on neural information processing systems (neurips 2017)* (pp. 3149–3157). Red Hook, NY, USA: Curran Associates, Inc. doi: https://doi.org/10.5555/3294996.3295074

Lan, G., Inan, H. A., Abdelnabi, S., Kulkarni, J., Wutschitz, L., Shokri, R., ... Sim, R. (2025). *Contextual integrity in llms via reasoning and reinforcement learning.* doi: https://doi.org/10.48550/arXiv.2506.04245

Lan, G., Zhang, S., Wang, T., Zhang, Y., Zhang, D., Wei, X., ... Brinton, C. G. (2025). *Mappo: Maximum a posteriori preference optimization with prior knowledge.* doi: https://doi.org/10.48550/arXiv.2507.21183

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). *Albert: A lite bert for self-supervised learning of language representations.* doi: https://doi.org/10.48550/arXiv.1909.11942

Malik, J. S., Qiao, H., Pang, G., & van den Hengel, A. (2025). Deep learning for hate speech detection: a comparative study. *International Journal of Data Science and Analytics*, *20*, 3055–3068. doi: https://doi.org/10.1007/s41060-024-00650-6

Mansur, Z., Omar, N., & Tiun, S. (2023). Twitter hate speech detection: A systematic review of methods, taxonomy analysis, challenges, and opportunities. *IEEE Access*, *11*, 16226–16249. doi: https://doi.org/10.1109/ACCESS.2023.3239375

Mathew, B., Saha, P., Yimam, S. M., Biemann, C., Goyal, P., & Mukherjee, A. (2021). Hatexplain: A benchmark dataset for explainable hate speech detection. In *The thirty-fifth aaai conference on artificial intelligence (aaai-21)* (pp. 14867–14875). doi: https://doi.org/10.1609/aaai.v35i17.17745

Rao Killi, C. B., Balakrishnan, N., & Rao, C. S. (2024). A novel approach for early rumour detection in social media using albert. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(3), 259–265. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/5248

Shah, S., & Patel, S. (2025). A comprehensive survey on fake news detection using machine learning. *Journal of Computer Science*, *21*(4), 982–990. doi: https://doi.org/10.3844/jcssp.2025.982.990

Tanvir, A. A., Mahir, E. M., Akhter, S., & Huq, M. R. (2019). Detecting fake news using machine learning and deep learning algorithms. In *2019 7th international conference on smart computing & communications (icscc)* (pp. 1–5). doi: https://doi.org/10.1109/ICSCC.2019.8843612

Tian, Y., Xu, S., Cao, Y., Wang, Z., & Wei, Z. (2025). An empirical comparison of machine learning and deep learning models for automated fake news detection. *Mathematics*, *13*(13), 2086. doi:

https://doi.org/10.3390/math13132086

Xu, S., Ding, Z., Wei, Z., Yang, C., Li, Y., Chen, X., & Wang, H. (2025). A comparative analysis of deep learning and machine learning approaches for spam identification on telegram. In *2025 6th international conference on computer communication and network security*.

Zhang, Y., Wang, Z., Ding, Z., Tian, Y., Dai, J., Shen, X., ... Cao, Y. (2025). Employing machine learning and deep learning models for mental illness detection. *Computation*, *13*(8), 186. doi: https://doi.org/10.3390/computation13080186