Learning and Individual Differences 44 (2015) 33–39

Contents lists available at ScienceDirect

Learning and Individual Differences

journal homepage: www.elsevier.com/locate/lindif

Dyadic instruction for middle school students: Liking promotes learning☆

Amy C. Hartl [a,*], Dawn DeLay [b], Brett Laursen [a], Jill Denner [c], Linda Werner [d], Shannon Campe [c], Eloy Ortiz [c]

[a] Department of Psychology, Florida Atlantic University, United States
[b] T. Denny Sanford School of Social and Family Dynamics, Arizona State University, United States
[c] Research Department, Education, Training, Research, United States
[d] Department of Computer Science, University of California, Santa Cruz, United States

ARTICLE INFO

Article history:
Received 15 April 2014
Received in revised form 18 February 2015
Accepted 4 November 2015

Keywords:
Friendship
Peer influence
Dyadic learning
Computer programming
Actor–partner interdependence model

ABSTRACT

This study examines whether friendship facilitates or hinders learning in a dyadic instructional setting. Working in 80 same-sex pairs, 160 (60 girls, 100 boys) middle school students ($M = 12.13$ years old) were taught a new computer programming language and programmed a game. Students spent 14 to 30 ($M = 22.7$) hours in a programming class. At the beginning and the end of the project, each participant separately completed (a) computer programming knowledge assessments and (b) questionnaires rating their affinity for their partner. Results support the proposition that liking promotes learning: Greater partner affinity predicted greater subsequent increases in computer programming knowledge for both partners. One partner's initial programming knowledge also positively predicted the other partner's subsequent partner affinity.

1. Introduction

Many teachers require students to work in dyads. Some teachers allow friends to work together; others do not. There are good reasons for each classroom strategy. Friends often collaborate enthusiastically, but there is concern they may be disruptive or digress from the task at hand (Jackson, Kutnick, & Kington, 2001; Zajac & Hartup, 1997). It is not clear from previous research if liking promotes learning in collaborative education. This is not just because research on the topic is limited but also because of the statistical obstacles posed by the nonindependent nature of dyadic data. The present study was designed to examine the effect of liking (or affinity for the partner) on learning in a collaborative computer programming environment. Analyses designed for nonindependent data assessed within-dyad, over time associations between partner affinity and the acquisition of computer skills.

Peers can be instrumental to learning. Peers are an important source of assistance, encouragement, and motivation, in that they model learning strategies and proffer tuition (Blatchford, Kutnick, Baines, & Galton, 2003; Kindermann, 2007; Tudge & Winterhoff, 1999). In collaborative settings, one peer usually has greater knowledge or experience than the other. Expert–novice interchanges are mutually beneficial because the novice learns content from the expert, and the expert's knowledge is strengthened by teaching (Azmitia, 1988; Howe, 2013; Rosenshine, Meister, & Chapman, 1996; Tudge & Rogoff, 1999; Vygotsky, 1978). According to Vygotsky (1978), when less skilled peers interact with more advanced peers, new skills are mastered when children work within their zone of proximal development, which represents the distance between actual and potential abilities. More advanced peers also cognitively benefit from collaboration. More specifically, in cooperative learning, each student within a group has specific responsibilities, actively participates in discussion, helps group members learn the material, assesses the progress of the group, and is held accountable for personally learning the material and helping the group learn the material (Johnson, Johnson, & Smith, 2007). The active, engaged nature of cooperative learning allows all group members to cognitively benefit, either through the instruction they offer others, which helps to consolidate and apply their knowledge or through the discussion with more skilled peers (Brown & Ciuffetelli, 2009). Finally, peers foster learning through discussion and negotiation (Azmitia & Perlmutter, 1989; Webb & Palincsar, 1996). It is clear that children profit from collaborative instructional opportunities; what is not clear is whether they profit more when they like their partner.

Conceptual views regarding the optimal partner for collaborative learning fall into two camps. Some argue that friends are advantageous; others argue that friends are a liability. The benefits side of the ledger

☆ This project was supported by a grant to Jill Denner and Linda Werner from the U.S. National Science Foundation (0909733). Brett Laursen also received support from the U.S. National Science Foundation (0923745) and the U.S. National Institute of Child Health and Human Development (HD068421).

* Corresponding author.
E-mail address: amy.hartl@live.com (A.C. Hartl).

http://dx.doi.org/10.1016/j.lindif.2015.11.002
1041-6080/© 2015 Elsevier Inc. All rights reserved.

includes views that closeness enhances the influence that peers wield over a variety of domains, including instruction and learning. Group work is most effective when there is trust, respect, mutuality, and equality among members (Kutnick & Colwell, 2010), characteristics usually present among friends. Friends work to maintain their relationship through open communication and successful conflict resolution (Laursen & Pursell, 2009), which provide a foundation for working together to solve problems and achieve a common goal. Considerable research supports claims that the closer peers are to one another, the more they influence each other's academic performance (Molloy, Gest, & Rulison, 2010). Many children and young adolescents working on collaborative tasks perform better when paired with friends than with nonfriends. This advantage extends to musical composition and communication (among girls) (MacDonald, Miell, & Mitchell, 2002; Miell & MacDonald, 2000), creativity box exploration and recall[1] (Newcomb & Brady, 1982), problem solving (Azmitia & Montgomery, 1993), and science reasoning (Kutnick & Kington, 2005). Collaboration with friends also promotes cognitive advances and task motivation (Fonzi, Schneider, Tani, & Tomada, 1997). Put simply, children prefer to work with friends, and there are many examples in which they benefit from doing so.

But not everyone agrees that children learn best in the company of friends. Some have argued the presence of friends can distract from learning and interfere with productivity (Zajac & Hartup, 1997). Working with a friend may relax partner expectations such that pressure to perform is reduced (Maldonado, Klemmer, & Pea, 2009). When friends are present, off-task behaviors can increase, which tends to limit on-task discussion (So & Brush, 2008). Together, these conspire to reduce effort invested in task completion. In one study of college undergraduates, ratings of friendship were negatively correlated with performance on a design project (Maldonado et al., 2009). Compared to nonfriends, friends proposed plans that were less well-developed and submitted final projects that were of poorer quality. The poor performance of friends was attributed to an aversion to criticize, pressure to agree, and time spent off-task.

Conclusions from prior research must be tempered by concerns about statistical analyses conducted on nonindependent dyadic data. Traditional parametric statistics are inappropriate for dyadic data, because correlated partner reports violate assumptions of independence, biasing error estimates (Kenny, 1996). Significance tests are compromised, which typically inflates Type II error. The problem is compounded in path analyses, because model misspecification biases parameter estimates.

Recent statistical advances overcome these difficulties. The Actor–Partner Interdependence Model (APIM: Kenny, Kashy, & Cook, 2006) partitions variance shared across partners on the same variables from variance that uniquely describes within- and between-partner associations. Modifications for longitudinal data specifically address over time influence between members of a dyad (Laursen, Popp, Burk, Kerr, & Stattin, 2008). A longitudinal APIM is akin to a residual change model, in that autoregressive effects describe the stability of a variable (Popp, Laursen, Kerr, Stattin, & Burk, 2008). By controlling for stability, the residual change in scores can be predicted. Applied to the present study, the APIM provides an unbiased estimate of the influence of liking or affinity on the acquisition of computer skills over the course of a dyadic computer programming class. One partner's initial perception of liking or affinity was used to predict improvement in computer programming knowledge of the same partner and of the other partner. Conversely, one partner's initial computer programming knowledge was used to predict changes in perceptions of liking or affinity for the partner.

Two research questions were addressed. Does liking promote learning? We tested two competing hypotheses. In keeping with the notion that friendship promotes motivation, we tested the hypothesis that greater partner affinity at the outset would anticipate greater increases in computer programming knowledge, for the self and for the partner. The alternative hypothesis, consistent with the notion that friends are often a distraction, holds that initial partner affinity would be inversely related to the rate of increase in computer programming knowledge. Does task knowledge promote liking? Little is known about whether children grow closer to those who excel in educational tasks, although one study suggests this might be the case (Tesser, Campbell, & Smith, 1984). If so, we would expect that initial computer programming knowledge would be positively associated with increases in partner affinity.

[1] Children explored a box that included interesting internal and external features. The features could be successfully manipulated by (1) only one child at a time, (2) one or two children at a time, or (3) a coordinated effort of two children.

2. Method

2.1. Participants

The final sample of 160 students (60 girls and 100 boys in a total of 80 same-sex dyads) included 61 6th graders, 49 7th graders, and 50 8th graders. Students ranged from 10 to 14 ($M = 12.13$, $SD = 1.00$) years old. Of this total, 64.4% ($n = 103$) lived with both biological parents, 18.1% ($n = 29$) lived with one biological parent, 3.1% ($n = 5$) lived in other households, and 14.4% ($n = 23$) did not report living arrangements. In the final sample, 45.6% ($n = 73$) described themselves as “White Caucasian”, 27.5% ($n = 44$) as “Mixed Race”, 13.8% ($n = 22$) as “Hispanic Latino”, 5.0% ($n = 8$) as “Asian Pacific Islander”, 3.1% ($n = 5$) as “Native American”, 1.9% ($n = 3$) as “African American”, and 3.1% ($n = 5$) did not report their race or ethnicity.

2.2. Measures

Participants separately completed the same assessments at the beginning of the project (pretest) and at the conclusion of the project (posttest).

Partner affinity was assessed with 6 items from the Friendship Quality Scale (Bukowski, Hoza, & Boivin, 1994), which measured perceptions of satisfaction with and affection for the partner (e.g., *I feel happy when I am with my partner*; *My partner is my friend*; *If my partner had to move away*, *I would miss him or her*; *When I do a good job at something my partner is happy for me*; *Sometimes my partner does things for me or makes me feel special*; *I can count on my partner for help*). Items were rated on a 1 (*not true*) to 5 (*really true*) scale. Scores were averaged across items. Internal reliability was good (Cronbach's $\alpha = .90$).

Computer programming knowledge was assessed with an 8-item measure of sample programming code created with the Alice interface similar to problems seen in introductory computer programming textbooks (Dann, Cooper, & Pausch, 2006). Items were created to measure skill in the programming environment and an understanding of programming code. Each item was designed to measure one or more of aspects of computational thinking, including algorithmic thinking, sequence of operations, and abstraction. Questions were similar to those found in computer science textbooks for children and novices. Questions tested students' declarative (e.g., facts about Alice) and procedural (e.g., ability to apply rules to solve problems) knowledge. Students were presented with screenshots of the programming interface (e.g., *What would happen if you were to play the above 3 lines of code*?). Students selected 1 of 5 possible response options. Each student received a score that indicated the number of correct responses (*Range* = 0–8). Internal reliability was acceptable (Cronbach's $\alpha = .76$).

2.3. Procedure

Participants were recruited from 8 classrooms and 4 extended learning programs in 7 public middle schools in 4 lower and middle class communities in Northern California. Schools were selected for participation if the technology teacher was interested in the project. In three