

Should Bao et al. (2020) be retracted?

Ian D. Gow

2022-10-13

Walker (2022) calls “for an investigation at the *Journal of Accounting Research* (JAR) into academic research misconduct” related to Bao et al. (2020). In this short note, I examine a somewhat different question: Should Bao et al. (2020) be retracted?

I argue that the current presentation of Bao et al. (2020) (the original paper plus an erratum) is apt to mislead regarding its key findings. As such, some form of retraction seems appropriate, even if only to provide (through a “retract and republish” approach) a research record that is both clear and free from known error.

I then identify a number of factors that seem relevant to evaluating the merits of providing the authors of Bao et al. (2020) the opportunity to republish if retraction were pursued.¹

1 Dramatis personae

Given the seemingly dramatic title, it is perhaps appropriate to start with a listing of the “characters” appearing in this note. In order of appearance (in the world, not in this paper) we have:

- The Original (BKLYZ0, Bao et al. 2015): Version submitted to JAR in 2015.
- The Publication (BKLYZ1, Bao et al. 2020): Version published in JAR in 2020.
- The Critique (W1, Walker 2021a): Published in *Econ Journal Watch*
- The Reply (BKLYZ2, Bao et al. 2021): Initial reply to W1 in *Econ Journal Watch*
- The Rejoinder (W2, Walker 2021b)
- The Erratum (BKLYZ3, Bao et al. 2022): Erratum published in JAR in 2022.
- The Call for an Investigation (W3, Walker 2022): Response to BKLYZ3.

¹The views expressed here are my own. I have no formal role in this process or association with any of the papers published in the *Journal of Accounting Research* or *Econ Journal Watch*. This note arises from observations I have made in preparing a chapter on prediction for a course book on accounting research. While I have sought feedback from a small number of people on the tone and content of this note, any errors herein are my own.

2 BKLYZ1

What is the contribution of BKLYZ1? Bao et al. (2020, 199) suggest that it is “a state-of-the-art fraud prediction model” where the fraud that is predicted is accounting fraud resulting in an Accounting and Auditing Enforcement Release (AAER) by the SEC.²

BKLYZ1 do indeed provide such a model. An analyst—whether an academic or a practitioner—can go to the [GitHub page associated with BKLYZ1](#), download the code and data, open Matlab, and generate the model ready for application to new data.³

But, even on the terms of BKLYZ1, this model has limitations. First, it doesn’t really detect accounting fraud in a general sense, so much as AAERs, a specific kind of fraud. Accounting fraud might not result in an AAER, either because it is never detected, or because it is detected but does not rise to the level that leads to an AAER, or even because the fraud is so severe that an AAER is somewhat irrelevant.

For example, with regard to the last category, it is not even clear that Enron, the public company at the heart of one of most notorious cases of accounting fraud this century, was the subject of an AAER. While the CEO (Jeffrey Skilling) and CFO (Andrew Fastow) of Enron ended up serving time in prison, there is no AAER related to either Skilling or Fastow (many AAERs relate to individuals). There is no AAER directed specifically at Enron, perhaps because it entered bankruptcy shortly after fraud was detected.⁴

Additionally, the BKLYZ1 sample “ends in 2008 because the regulators reduced the enforcement of accounting fraud starting from around 2009, increasing the possibility that many accounting fraud cases remain undetected for the post-2008 period” (Bao et al. 2020, 203–4). In other words, the specific outcome (AAERs) that the BKLYZ1 model is designed to predict becomes simply too difficult to predict after 2008. This means that the model is only useful for “predicting” AAERs before 2009. Obviously no practitioner would find such a model useful and even academics probably have little use for a model whose utility ended with the financial crisis of 2008.

But even if the BKLYZ1 model itself is not useful for any identifiable purpose, perhaps the contribution of BKLYZ1 is in showing that a model based on ensemble learning (“one of the

²See the SEC [website](#) for details.

³In some ways, the requirement for Matlab is unfortunate, as Matlab is proprietary software and offers less transparency than an implementation using one of the open-source alternatives, such as Python or R, would. Unfortunately there is no implementation of RUSBoost in the popular Python library, `scikit-learn`, though this library does have an implementation of AdaBoost. There is also no well-documented implementation of RUSBoost in R, though there are several implementations of the AdaBoost in R. Fortunately, it is possible to implement RUSBoost in R, and I include an implementation in the R package `farr` that I created as a complement to the course book found [here](#). A chapter using RUSBoost will be forthcoming in the near future.

⁴The [one AAER](#) in the Bao et al. (2020) sample connected to Enron actually covers the order for Citigroup to pay an amount in a settlement arising because “Citigroup assisted [Enron and Dynegy] in enhancing artificially their financial presentations through a series of complex structured transactions ... to allow those companies to report proceeds of financings as cash from operating activities”.

most powerful machine learning methods”) “outperforms ... by a large margin” approaches commonly used in prior accounting research, including models based on logistic regression. Of course, this would not be a contribution to the vast literature in statistical learning, as any practitioner in that field is unlikely to be surprised by a general result covered in an introductory textbook.⁵

Nor would we view the contribution of BKLYZ1 to be in demonstrating the superiority of the specific ensemble method used in the paper (“RUSBoost”) over alternative ensemble methods. First, BKLYZ1 do not evaluate RUSBoost relative to such other methods, including the Adaboost approach on which RUSBoost is based. Second, earlier research has already provided evidence on this point (e.g., Seiffert et al. 2008).

Rather the contribution of BKLYZ1 seems more specific to the setting of accounting fraud. Bao et al. (2020, 204) summarize their results:

The average AUC and the average NDCG@k for the ensemble learning model are 0.725 and 0.049, respectively, representing a performance increase of 7.9% and 75%, respectively, relative to the performance of the better benchmark model, the Dechow et al. model [based on logistic regression]. These performance differences are also economically significant: Using the NDCG@k approach (where $k = 1\%$), our best model, the ensemble learning model, identified a total of 16 fraud cases in the test period 2003–08 [versus] 9 for the Dechow et al. model.

Assuming that the results of BKLYZ1 generalize from the prediction of AAERs in 2003–2008 to prediction of accounting fraud more broadly, we might conclude that BKLYZ1 provides the helpful result that a model based on RUSBoost can provide superior performance in predicting accounting fraud than a model based on logistic regression. Presumably this was the central contribution of BKLYZ1 that led to its publication in an accounting outlet as prestigious as the *Journal of Accounting Research*.

3 BKLYZ3

Unfortunately, we need to update the central finding of BKLYZ1 in light of The Erratum (BKLYZ3). BKLYZ3 corrects a “coding error” in BKLYZ1 identified by W1. The updated results are found in Panel B of Table 1 of BKLYZ3. From BKLYZ3 we learn that the summary of results quoted above should instead read.

The average AUC and the average NDCG@k for the ensemble learning model are 0.7228 and 0.0237, respectively, representing a performance increase of 7.7% and

⁵The general result here being that statistical learning methods such as ensemble learning usually improve out-of-sample prediction performance relative to models—such as logistic regression—that tend to *overfit* the data used to train them.

performance decrease of 13.2%, respectively, relative to the performance of the better benchmark model, the Dechow et al. model [based on logistic regression]. These performance differences are also economically significant: Using the NDCG@k approach (where $k = 1\%$), our best model, the ensemble learning model, identified a total of 10 fraud cases in the test period 2003–08 [versus] 8 for the Dechow et al. model.

So, based on the criteria used in BKLYZ1, correction of the “coding error” in that paper overturns the central result of BKLYZ1. So what seems to be the sole basis for publishing BKLYZ1 is no longer true.

Indeed, W3 (Walker 2022, 192) suggests that the second sentence may need an additional correction to read:

These performance differences are also economically significant: Using the NDCG@k approach (where $k = 1\%$), our best model, the ensemble learning model, identified a total of 8 fraud cases in the test period 2003–08 [versus] 8 for the Dechow et al. model.

A reader who stumbled upon BKLYZ1 and then checked BKLYZ3 could easily be forgiven for missing how the primary result of BKLYZ1 is undermined by the correction in BKLYZ3. BKLYZ3 arguably obfuscates this impact by emphasizing alternative test periods, placing renewed emphasis on AUC (the metric with weaker results in BKLYZ1), introducing novel results using NDCG@k at cut-offs not considered in BKLYZ1, and providing a rather beside-the-point discussion of an “alternative approach to coding serial fraud” that does not actually provide an alternative approach.⁶

4 Are their grounds for retraction?

The Committee on Publication Ethics (COPE) provides guidance “intended to advise editors and publishers on expected practices when considering whether a retraction is appropriate, and how to issue a retraction.”⁷ In light of BKLYZ3, “editors should consider retracting a publication if ... they have clear evidence that the findings are unreliable ... as a result of major error (eg, miscalculation or experimental error).” Given that BKLYZ3 provides evidence of an error and, based on our discussion above, this error goes to the central result of the BKLYZ1, retraction seems to be an appropriate response.

⁶AAER dates are easily obtained from the SEC website, and included with the `farr` R package (see [here](#)). My own analysis suggests that AAERs are never released prior to the last affected period, so AAERs that affect test years are always in the “future” relative to that test year, and should never be coded as anything other than zero in the training sample.

⁷See p.2 of [Retraction Guidelines](#).

But note that the COPE guidelines [p. 8] provide the option of “retract and republish”: “journals may wish to work with authors to concurrently retract an article that was found to be fundamentally flawed while simultaneously publishing a linked and corrected version of the work. This strategy ... may provide an opportunity for journals and authors to transparently correct the literature when a simple correction cannot sufficiently address the flaws of the original article.”

Note that, while many grounds for retraction involve evidence of academic misconduct, there is no requirement that misconduct be shown for a retraction to occur. Indeed, it seems that the “retract and republish” option that plausibly has merit in the present case is predicated on the *absence* of misconduct.

4.1 Is there evidence of misconduct?

W3 claims to “make the case that there is evidence of academic misconduct and make the recommendation that the *Journal of Accounting Research* launch a full and independent investigation into the matter.” I cannot imagine that the editors of the *Journal of Accounting Research* would be keen to get into questions of academic misconduct, given the implications of any such finding. Instead, it seems more relevant to focus on issues pertinent to what is published in JAR.

While I argue that a better presentation of the research record requires some kind of retraction, the COPE guidelines cited above appear to afford some latitude as to whether a journal will “in some instances ... wish to work with authors to concurrently retract an article that was found to be fundamentally flawed while simultaneously publishing a linked and corrected version of the work.” In other words, the *Journal of Accounting Research* arguably enjoys wide discretion over whether to “republish” a corrected version of BKLYZ1.

The remainder of this note collects some information that I conjecture the *Journal of Accounting Research* might consider in reaching its decision on the best course of action in response to the call from W3.

5 Some observations

5.1 Test period

One thing that BKLYZ1 is very clear on is that the test period is 2003–2008. The main results (“performance increase of 7.9% and 75%”) of that paper are all based on this test period.

Yet the erratum [BKLYZ3, p. 1636] mysteriously seems to emphasize 2003–2005 as the test period: “Using NDCG@k as a performance measure RUSBoost ... continues the dominate the performance of the other models for the test period 2003–2005.” The published erratum

[p. 1636] even suggests that BKLYZ1 “argued that this test period was the cleanest”. Yet this is simply an impossible reading of BKLYZ1. As a reader, one needs to be confident that the “best test period” is not simply the test period that delivers the most favorable “out-of-sample” performance for RUSBoost.

At this point, a careful reader might point out that 2003–2005 was actually the test period used in BKLYZ0. Hopefully, BKLYZ would not be the ones to point this out, because this choice of test period was justified in BKLYZ0 on the following two bases. First, BKLYZ0 state [p. 4] that “the SEC’s Accounting and Auditing Enforcement Releases (AAERs) available to us end in September 2010”. Second, “there is an average of five-year gap between a fraud occurrence and the AAER publication date.”

As the sample period in BKLYZ1 includes AAERs that extend to 2014, the first item seems to suggest that (on the logic of BKLYZ0 itself) with the test period could be updated to from 2003–2005 to 2003–2009 (i.e., adding four years).

But it is important to note that BKLYZ0 included in their training sample frauds that were also in the test periods, even though the AAER publication dates would have been after the test periods in question.⁸

If the second basis were maintained, but the issues of “serial fraud” addressed, then by the logic of BKLYZ1, the “gap” of two years used in BKLYZ1 would have to be five years, and thus the feasible test sample could not *begin* until 2006. BKLYZ1 [p. 209] “require a gap of 24 months between the financial results announcement of the last training year and the results announcement of a test year ... because Dyck, Morse, and Zingales (2010) find that it takes approximately 24 months, on average, for the initial disclosure of the fraud.”⁹

5.2 The “coding error”

BKLYZ3 states [p. 1635] that [W1 and W2] “identified an error in the program codes [sic] of [BKLYZ1](#) posted on Github that led to an overstatement of model performance metrics. This erratum corrects this error ...” It is difficult to disagree with W3’s claim that this statement is false. There is nothing in the code posted on GitHub that created this error, instead the “error” was in data used by that code.

W3 claims that “to this date, the authors have offered no explanation as to why they did what they did” in recoding certain frauds to have different identifiers. This seems correct. In BKLYZ2, BKLYZ provide what may be best described as a non-explanation for what they did.

⁸That this is the case is implicit in footnote 10 to BKLYZ1 and the discussion under “serial fraud” in BKLYZ1, which is not found in BKLYZ0.

⁹While this shorter period is definitely convenient for BKLYZ1, it seems less convenient that there is no evidence of the claim in Dyck, Morse, and Zingales (2010) itself. Perhaps the BKLYZ1 authors obtained underlying data from the authors of Dyck, Morse, and Zingales (2010).

The example in Figure 1 of BKLYZ2 illustrates the “coding error” made in BKLYZ1 if there are missing items for the affected firm in 2004. If such missing items exist, firm-years in 2001 through 2003 would be given a different fraud ID from the 2005 firm-year (in such cases BKLYZ appended the character “1” to the fraud ID for 2001 through 2003, and the character “2” to the fraud ID for 2005). Because the fraud IDs for 2001–2003 no longer appear in the test year (2005), they are not recoded as zero. In BKLYZ2, the practice of *not* recoding these frauds in this way is described as “Walker’s approach” even though (as W2 points out), this is not so much Walker’s approach as the approach described in BKLYZ1.

Evidenced of this approach to recoding frauds is not legitimate is provided by the fact that doing it that way has been characterized as a “coding error” in BKLYZ3.

But relabelling “serial frauds” also does not make sense in that it is recoding a fraud as 1 in 2003 so that a “prediction” of the same underlying fraud can be made in 2005 even though, by the terms of the example itself (Figure 1 of BKLYZ), the SEC does not release an AAER until 2007.

That training models using data on frauds that are not released until after the test period is problematic seems obvious. And it seems clear from BKLYZ1’s discussion of “serial fraud” that the authors were well aware of these issues. In fact, they seem aware of the underlying issue even in BKLYZ0. As discussed above, the BKLYZ0 “sample ends in 2005 because ... a significant portion of the accounting frauds that occurred over 2006–2010 are likely still unreported by the AAERs as of the end of 2010.”

One response the authors might have is that there might be a “key event [that] reveals 2001–2003 fraud labels” before 2004 (and before the AAER publication date in 2007). But this implies a completely different prediction problem from that studied in BKLYZ1 (or BKLYZ0), which identifies frauds as AAER events. The only reliable “key event” that identifies an AAER is the release of an AAER. As discussed above, disclosed accounting fraud might not result in AAERs for a number of reasons.

Saying that sometimes information is released that suggests a high likelihood of a future AAER event transforms the prediction problem from one about predicting confirmed AAERs using financial statement features into one about predicting confirmed AAERs using financial statement features and also some *unidentified* information about possible future AAERs.

Even if we expand the information set to include the “maybe-future-AAERs” as is done in the “not Walker’s approach” depicted in Figure 1 of BKLYZ2, there is no rationale provided as to why missing values for some items on Compustat in 2004 should be assumed to precipitate a fraud revelation event before 2004. In fact, it seems hard to conceive of one.

And this “coding error” is surely not something that happened by mistake. The strident defence of their approach provided in BKLYZ2 suggests that the authors did this consciously, and only in BKLYZ3 did they suggest it was a “coding error”. A reproduction of the “coding error” is included in an appendix to this note and it seems difficult to see how this “coding error” could have been made inadvertently.

At the very least, I think the authors should be required to provide more information of how the “coding error” was implemented so that the editors can assess the likelihood that it was indeed a “coding error” (as it is framed in BKLYZ3) and not a deliberate research design choice (as it is framed in BKLYZ2). If it seems that it was a deliberate research design choice, then I think the authors need to explain the rationale for it and also the rationale for disassembling its existence in the code posted to GitHub upon publication of BKLYZ1.

One possible response by the BKLYZ team might be that they have more than complied with the JAR data policy in effect when they submitted their paper and therefore do not need to account for the “coding error” beyond the code they have already provided.

I do not think this kind of response would be helpful in this case. There appears to be no data policy at *The Accounting Review* (see discussion [here](#)), but this did not prevent the retraction of Bird and Karolyi (2017), where the journal stated “the authors were unable to provide the original data and code requested by the publisher” to support an assertion made in the paper and therefore retracted the paper.

Given the “coding error” of BKLYZ3 appears to have been a conscious decision (see BKLYZ2), I think that the onus is on the authors of BKLYZ1 to show that the “coding error” was made in good faith.

5.3 Meta-parameters

In BKLYZ2, the authors suggested that W1 was flawed because he “did not recalibrate the most important parameter of RUSBoost, number of trees, after changing the fraud training samples using his approach [i.e., the approach described in BKLYZ1].” This is somewhat understandable, as BKLYZ1 does not describe the process of calibrating these meta-parameters in the first place. Indeed, there are several meta-parameters used in BKLYZ1: `number of trees`, `MinLeafSize`, `LearnRate`, and `RatioToSmallest`. It is critical that such meta-parameters be fixed using the training and validation data prior to evaluating model performance against test data, lest these parameters be selected based on test performance, thus overstating the predictive value of the model.

In this regard, it is somewhat concerning that the number of trees parameter that is selected for the BKLYZ1 specification is 1,000 according to W3, not the 3,000 used in BKLYZ1.¹⁰ This creates the unfortunate impression that the idea of selecting meta-parameters in a transparent fashion using only validation data emerged only after observing a decline in test performance upon switching to “Walker’s approach” in preparing the erratum.

¹⁰In my own analysis, I found that 900 trees maximized performance in the validation sample, which is very close to the value found in W3.

5.4 Other data issues

While it is possible to explain changes in the code between BKLYZ1 and BKLYZ3 using GitHub, some concerns remain. For example, as detailed [here](#), the two data files provided are not consistent. While the PDF-rendered SAS code suggests that one data file depends on the other, there are observations on AAERs not found in the former.

5.5 How unusual are the issues in BKLYZ1?

Some readers may be surprised to learn that the core results of a published paper can disappear when a “coding error” is detected and corrected. I am not surprised. Having replicated many papers, I conjecture that the kinds of issues observed with BKLYZ1 are commonplace. Papers that say one thing, but do another (most papers using “regression discontinuity designs” fit [here](#)). Papers with genuine coding errors. Papers with results that are very sensitive to “design choices” that are difficult to rationalize (see [here](#) and [here](#)). If such issues abound, then the merit of singling out BKLYZ1 seems to be low.

Another factor that seems relevant is JAR’s data policy. While not perfect, arguably JAR’s policy was critical in helping to unearth the issues not only in BKLYZ1, but in the replications I refer to above.¹¹ If BKLYZ1 had been accompanied by a very perfunctory effort to comply with the data policy (e.g., “here is the list of CIKs for the fraud firms in our sample”), then it would not have been possible to detect the issue raised by W1 and corrected in BKLYZ3.

6 Appendix: Reproducing the coding error

In this appendix, I recreate the “coding error” in Bao et al. (2020) that was corrected by Bao et al. (2022).

To do this, I will use the `tidyverse` package.¹²

```
library(tidyverse)
```

6.1 The source data sets

[PDF-rendered SAS course code](#) supplied by BKLYZ suggests that the final data set used in Bao et al. (2020) was constructed by merging data on AAERs (`aaer_firm_year`) with data

¹¹We should not infer that the situation is better at journals without a data-sharing policy. If anything, we might expect it to be worse.

¹²Install this using the command `install.packages("tidyverse")` in R, if necessary.

on raw Compustat variables (`compustatindustrial7815`). To reproduce the “coding error”, we need to retrace the process of merging these two tables.

While BKLYZ provide a file `AAER_firm_year.csv`, it is easy to show that this was not the data file used to create the Bao et al. (2020) data set.¹³

As such, we use the “final” data set used by Bao et al. (2020) and reconstruct the relevant portions of the source data sets from that.

```
jar_data <-
  read_csv(paste0("https://raw.githubusercontent.com/JarFraud/",
                  "FraudDetection/master/",
                  "data_FraudDetection_JAR2020.csv"),
           col_types = "d") %>%
  mutate(gvkey = str_pad(gvkey, 6, side = "left", pad = "0"),
         fyear = as.integer(fyear),
         p_aaer = as.character(p_aaer))
```

We first construct the original `aaer_firm_year` data set by filling in any gaps in firm-years for an AAER found in `jar_data`.

```
aaer_firm_year <-
  jar_data %>%
  filter(!is.na(p_aaer)) %>%
  group_by(p_aaer, gvkey) %>%
  summarize(min_year = min(fyear), max_year = max(fyear),
            .groups = 'drop') %>%
  rowwise() %>%
  mutate(fyear = list(seq(min_year, max_year, by = 1))) %>%
  unnest(fyear) %>%
  select(gvkey, fyear, p_aaer)
```

As we can see here, `aaer_firm_year` is the panel data set of firm-years affected by AAERs.

```
head(aaer_firm_year)
```

```
# A tibble: 6 x 3
  gvkey  fyear p_aaer
<chr>  <dbl> <chr>
1 021110  1992 1033
2 008496  1992 1037
```

¹³See [here](#) for details. In short, the final data set used in Bao et al. (2020) contains AAERs not found in `AAER_firm_year.csv`.

```

3 008496 1993 1037
4 028140 1993 1044
5 012455 1994 1047
6 025927 1993 1053

```

6.2 Firm-years with Compustat features

We next construct the data set `comp_firm_years`, which represents the firm-years in the Bao et al. (2020) sample. These will be missing some of the firm-years in `aaer_firm_year` because of missing items on Compustat.

```

comp_firm_years <-
  jar_data %>%
  mutate(missing = 0) %>%
  select(fyear, gvkey, missing)

```

6.3 The “coding error”

Now that we have the two data sets `aaer_firm_year` and `comp_firm_years`, we can reproduce the “coding error” from Bao et al. (2020).

The first step is to merge `aaer_firm_year` with `comp_firm_years` using the `left_join` function so that all observations on `aaer_firm_year` are retained even if there is no match on `comp_firm_years`. The cases where there is no match on `comp_firm_years` are indicated by the variable `missing`.

```

aaer_merged <-
  aaer_firm_year %>%
  left_join(comp_firm_years, by = c("gvkey", "fyear")) %>%
  mutate(missing = coalesce(missing, 1)) %>%
  select(gvkey, fyear, p_aaer, missing)

head(aaer_merged)

```

```

# A tibble: 6 x 4
  gvkey  fyear p_aaer missing
<chr>  <dbl> <chr>    <dbl>
1 021110 1992 1033      0
2 008496 1992 1037      0
3 008496 1993 1037      0
4 028140 1993 1044      0

```

5	012455	1994	1047	0
6	025927	1993	1053	0

Now we can recode AAERs following Bao et al. (2020). We do this by calculating `sum_gap`, a running sum of the indicator variable for one plus the number of gaps in the sample for a given AAER.¹⁴ This will start at 1 and increase to 2 after a “gap”. We then create the variable `new_p_aaer` by combining the original AAER identifier (`p_aaer`) with `sum_gap`.

```
aaer_sum_gap <-
  aaer_merged %>%
  group_by(gvkey, p_aaer) %>%
  arrange(fyear) %>%
  mutate(lag_missing = coalesce(lag(missing), 0),
         sum_gap = cumsum(lag_missing & !missing) + 1,
         new_p_aaer = paste0(p_aaer,
                             as.character(sum_gap))) %>%
  select(gvkey, fyear, p_aaer, new_p_aaer) %>%
  ungroup()
```

If successful, my code reproduces the “coding error” in Bao et al. (2020) in about 14 lines of code.¹⁵

6.4 Verifying the reproduction of the “coding error”

To check that we have successfully reproduced the “coding error” corrected by Bao et al. (2022), we can reproduce Figure 2 of Walker (2021a), which we do with the following code.

```
walker_fig_2 <-
  aaer_sum_gap %>%
  group_by(p_aaer) %>%
  filter(n_distinct(new_p_aaer) > 1) %>%
  inner_join(comp_firm_years, by = c("gvkey", "fyear")) %>%
  pivot_wider(names_from = "fyear", id_cols = "p_aaer",
              values_from = "new_p_aaer") %>%
  ungroup() %>%
  arrange(as.integer(p_aaer)) %>%
  mutate(across(`1991`:`2014`, ~ coalesce(., "")))
```

For reasons of space, we only include a portion of the table.

¹⁴A “gap” is indicated by `lag_missing` being 1 and `missing` being zero.

¹⁵The exact number depends on how one counts a line.

```
walker_fig_2 %>%
  select(p_aaer, `1995`:`2004`) %>%
  knitr::kable()
```

p_aaer	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004
857										
1542			15421	15421		15422				
1839				18391	18391	18391		18392		
2472				24721	24721	24721	24721		24722	24722
2504						25041	25041		25042	25042
2591						25911	25911		25912	25912
2754	27541	27541		27542	27542	27542	27542	27542	27542	27542
2894						28941		28942	28942	
2937				29371						29372
2949					29491					29492
3022					30221			30222		
3045			30451	30451	30451		30452			
3156									31561	
3217		32171	32171			32172	32172	32172	32172	32172
3909										
3996										

Careful comparison of my data with Figure 2 of Walker (2021a) suggests that I have almost perfectly reproduced the “coding error” of Bao et al. (2020). The one exception is that my table omits the AAER with `p_aaer` of 2957.

Examining the underlying data, it seems that the issue here is the presence of multiple AAERs for the related firm for 2000 and 2001.

```
aaer_firm_year %>%
  filter(gvkey == "064630") %>%
  arrange(fyear)
```

```
# A tibble: 8 x 3
  gvkey  fyear p_aaer
<chr>  <dbl> <chr>
1 064630  1997 2957
2 064630  1998 2957
3 064630  1999 2957
4 064630  2000 2259
```

```

5 064630 2000 2957
6 064630 2001 2259
7 064630 2001 2957
8 064630 2002 2957

```

Returning to the SAS code supplied with Bao et al. (2020), we see the following lines:

```

PROC SORT DATA=temp nodupkey;
BY gvkey fyear; RUN;

```

This code would have the effect of (essentially randomly) deleting data on one AAER for any firm-year where two AAERs apply. It is unclear whether the BKLYZ team would characterize this second basis for recoding AAERs as a research design choice (as the coding error was arguably characterized in Bao et al. 2021) or as an error like the “coding error” replicated above.

The use of PROC SORT `nodupkey` is common in accounting research, but in general this is a problematic practice.¹⁶

References

- Bao, Yang, Bin Ke, Bin Li, Y. Julia Yu, and Jie Zhang. 2015. “Detecting Accounting Frauds in Publicly Traded U.S. Firms: New Perspective and New Method.” https://www.rieb.kobe-u.ac.jp/tjar/conference/6th/CA3_YingriJuliaYU.pdf.
- . 2020. “Detecting Accounting Fraud in Publicly Traded U.S. Firms Using a Machine Learning Approach.” *Journal of Accounting Research* 58 (1): 199–235. <https://doi.org/10.1111/1475-679X.12292>.
- . 2021. “A Response to ‘Critique of an Article on Machine Learning in the Detection of Accounting Fraud.’” *Econ Journal Watch* 18 (1): 71–78. <https://econjwatch.org/articles/a-response-to-critique-of-an-article-on-machine-learning-in-the-detection-of-accounting-fraud>.
- . 2022. “Erratum.” *Journal of Accounting Research* 60 (4): 1635–46. <https://doi.org/10.1111/1475-679X.12454>.
- Bird, Andrew, and Stephen A. Karolyi. 2017. “Governance and Taxes: Evidence from Regression Discontinuity.” *The Accounting Review* 92 (1): 29–50. <https://doi.org/10.2308/1558-7967-92.1.000>.
- Dyck, Alexander, Adair Morse, and Luigi Zingales. 2010. “Who Blows the Whistle on Corporate Fraud?” *The Journal of Finance* 65 (6): 2213–53. <https://doi.org/10.1111/j.1540-6261.2010.01614.x>.

¹⁶Issues associated with this practice are explored in discussion questions [here](#).

- Seiffert, Chris, Taghi M. Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. 2008. “RUSBoost: Improving Classification Performance When Training Data Is Skewed.” In *2008 19th International Conference on Pattern Recognition*, 1–4.
- Walker, Stephen. 2021a. “Critique of an Article on Machine Learning in the Detection of Accounting Fraud.” *Econ Journal Watch* 18 (2): 61. <https://econjwatch.org/articles/critique-of-an-article-on-machine-learning-in-the-detection-of-accounting-fraud>.
- . 2021b. “Rejoinder to the Critique of an Article on Machine Learning in the Detection of Accounting Fraud.” *Econ Journal Watch* 18 (2): 230. <https://econjwatch.org/articles/rejoinder-to-the-critique-of-an-article-on-machine-learning-in-the-detection-of-accounting-fraud>.
- . 2022. “Erroneous Erratum to Accounting Fraud Article.” *Econ Journal Watch* 19 (2): 190–203. <https://econjwatch.org/articles/erroneous-erratum-to-accounting-fraud-article>.