Terrorist Attacks, Cultural Incidents, and the Vote for Radical Parties: Analyzing Text from Twitter

Online Appendices

A Appendix: Data

Landmarks

The final goal of our data collection process was to obtain a set of Twitter users allocated within German constituencies. The data section in the main part explained how we chose cities and towns in constituencies. For each of these towns, we manually obtained a set of landmarks Twitter accounts for which the key requirement is a high chance of being followed by local residents, but at the same time a low probability to be followed by non-residents. Remember here that with resident we include constituency area surrounding a city or town. For each city, the procedure started by searching for that specific town name in the Twitter search bar. Then, each result was individually assessed. We used direct and location search of the town name to find accounts that could represent landmarks. We employed a conservative stance, meaning that if an account gave reason for doubt it was not considered. Common sets of accounts which occurred frequently across cities and towns were:

• accounts by the city administration, which usually provide information onlocal events, regulations, or conditions (e.g. weather stations, traffic news)

• accounts by the local law enforcement agencies, fire departments, and other emergency facilities

• local business accounts of shops (but not larger shopping centers), markets, cinemas, barber shops, coffee shops, restaurants, bars, and similar facilities.

• leisure activities, such as bicycle, hiking, or running groups, as well as organized local sport activities (e.g. gymnastic clubs) excluding sport teams with a potentially wide-spread fan bases

• local news media and radio stations. Here, a more critical assessment was applied by including information provided on websites into the decision process. For example, small-scale local newspaper usually provide information on the towns in which they are sold. It was assessed whether this set of towns lay within (but not beyond) the constituency it was supposed to cover. A similar case is represented by local radio station, which tend to provide information on their local coverage area.

In order to be included, the user had to follow at least 3 landmark accounts. As described in the main text, this strategy could potentially lead to the inclusion of users which follow such accounts from several cities, for instance if an active Twitter user commuted over far distances or moved from one place to another. In case our allocation rule located a user in more than one constituency, the user was dropped from the sample.

Users

Employing the above described strategy lead first to a set of 982,358 users who follow at least one landmark. 234,468 of those were eliminated because the followed landmarks in more than one constituency. Of the remaining users, 447,401 users follow only one landmark and 111,071 follow only two landmarks, resulting in another sample reduction of 189,418 users who follow less than thee landmarks.

Predicted Users Characteristics

We use a Machine Learning algorithm described in Wang et al. (2019) and implemented in the m3inference library available in Python. The algorithm uses a multimodal deep neural architecture for joint classification of age, gender, and organization-status of Twitter users by looking at their username, screen name, provided biography, and profile image. We use this pre-trained deep neural model to predict the age, gender, and organization status of the users in our sample.

When the probability of the model prediction is below 0.75, we consider the prediction as missing. For instance, if the model predicts that a user is a man with 0.70 probability, we do not assign the user to any gender as the model prediction is not that certain. In the case of gender, we were not able to predict it with reasonable certainty for just 0.06% of our users. We find that 73% of our users are male.

Ethical Considerations

The use of algorithms to predict demographic traits of individuals rests on a sequence of explicit or implicit choices about the categories to use for classification and the final use of the classification. Each of these choices is not neutral, but substantive, and can potentially have real consequences on the human subjects comprising the sample and society at large. When designing or using algorithms for classification of humans, researchers should be compelled to consider the implications of the procedure they are adopting. Required considerations on the researcher's side include: (i) what is the intended purpose of the classification data produced, and how they could be used by third parties, (ii) how categories for demographic classification are defined, especially those that are at the core of individual identity like gender, ethnicity, or race (Scheuerman et al. 2020).

In this paper we use a pre-trained algorithm for age and gender classification based on publicly available information from Twitter accounts (Wang et al. 2019). The algorithm does not predict ethnicity or race, and it thus avoids some of the adverse consequences of ethnic classification, for instance the possible reinforcement of stereotypes about members of specific groups, or the provision of tools for minority targeting (Scheuerman et al. 2020). However, the algorithm does predict individual gender. We note that this feature may be potentially problematic in two ways. First, the gender classification adopted by Wang et al. (2019) is binary, therefore excluding by construction non-binary gender identification. Second, the classification exercise is conducted having in mind a specific sub-population, that of supporters of a political party. With respect to the latter issue, our approach is not meant to, nor could, suggest that a certain gender is more likely to support a radical right party. The classification only derives a predicted gender distribution, which we compare to a known distribution within the sub-population of supporters, taken from official data. With respect to the former issue, we acknowledge that the adoption of a binary definition of gender is a restrictive feature of the algorithm. Given the purpose of our analysis, namely characterizing sample selection relative to externally provided benchmarks, we are bound to adopt demographic categories that match those in the official data we use. We are conscious that the choice to adopt an existing classification is not a neutral decision, as it can reinforce and legitimize discriminatory practices. We advocate for the adoption of more comprehensive gender, racial and ethnic categories among public administrations, in order to minimize the problems that classification by authorities can pose to the lives of individuals.

B Appendix: Tweets and Content

Parties' Tweets

In the following we provide the details on how the word comparison graphs are created. We first compute the following log-odds-ratio for each word[1] w in the tweets of a party:

$$log\,log\,or_w = log\,log\,\left[\frac{f_{i,w}}{1-f_{i,w}}\right] - log\,log\,\left[\frac{f_{j,w}}{1-f_{j,w}}\right]$$

where $f_{i,w}$ is the frequency of a word w in document i.[2] This ratio identifies words which are most likely to appear in a party's tweets and at the same time least likely in another party's tweets, thus allowing us to identify what one party is most concerned about but the other one is not. We then rank them from highest to lowest and take the raw difference of occurrences for each word between documents and standardize it (Figure 2a). This allows reading one unit in the graph as one standard deviation of differences in occurrences. We further categorize the resulting list. Some of these words naturally occur mainly in one of the parties' tweets but not in another, such as the names of politicians and party specific congresses. Since we are mainly interested in identifying the words with political relevance we manually categorize each word, such that we know whether it is about a political topic or about something else, like the name of a politician, the reference to an event or non-identifiable junk.

After obtaining a categorized list still ordered by the ratio shown above (whose usefulness for a ranking were discussed), we plot the standardized raw difference (which we expect to be easier to read compared to a log-odds ratio) for the top and bottom 15 words of our ranking.[3]

Latent Dirichlet Allocation (LDA)

Using the pre-processed text that we use for the doc2vec model, we fit a guided LDA model (Jagarlamudi, Daumé, Udupa 2012) on our entire corpus of parties' and public's tweets. We explore models with different possible number of topics with 16 topics: immigration, Islam, elections, soccer, world politics, education, economy, arts (music and film), cities, digital, spare time, house, mobility, social networks, information, and interviews. We report in Table B.1 the words that we use as seeds for each topic. We classify a tweet as discussing a certain topic if the model returns a probability higher than 0.90. We then randomly sample 900 tweets and use a human coder to label them. We then evaluate the precision, recall, and F1-score of the LDA predictions of each topic against this human coding. The overall precision is 0.83, recall 0.89, and F1-score 0.85.

---

[1] With the term ”word”, we actually mean a ”token” after pre-processing the tweets, as explained below in appendix section C.

[2] We use the log normalization to make the odds-ratios symmetric across documents.

[3] All phrases and words in Figure 2 are translated from German into English.

Newspaper Analysis: Salience and Sentiment of Immigration and Islam

All newspaper articles in this study were filtered and collected from the digital news database Factiva. We focus on the six most circulated newspapers in Germany. They were chosen based on print circulation, as reported by IVW (2015). The newspapers (written in their Factiva name) are: "BILD - All sources", "Die Welt" with "Weltam Sonntag" and "Welt Online", "Frankfurter Allgemeine Zeitung", "Süddeutsche Zeitung" with "Süddeutsche Zeitung Online", "Handelsblatt", "Westdeutsche Allgemeine Zeitung".

In order to understand whether the increasing interest in Islam and immigration is driven by natural conversation or by media, we collect the weekly number of German newspaper articles that contain keywords related to immigration ("*migration*","*wander*", "*flüchtling*", and "*asyl*") and Islam ("*Islam*" and "*mus-lim*").

For our sentiment analysis based on article text, we furthermore filter Factiva on subject being "Political/General News", Region being Germany, language being German, and date range being September 04, 2015 to September 24, 2017. Then, we recruit 4 German-speaker annotators on Upwork to manually annotate a random sample of 929 newspaper articles. Each of them, classified the sentiment of an article in three classes: negative (-1), neutral (0), and positive (1). We train different classifiers with the `sklearn` library in Python and find that the best one is a Ridge classifier. This classifier reaches an accuracy score of 0.91. We then use this classifier to predict the sentiment of the remaining articles. We then evaluate the precision, recall, and F1-score of the predictions against further 100 articles annotated by other human coders. Precision is 0.98, recall 0.96,and F1-score 0.95, increasing confidence in the generalizability of the classifier.

Sentiment towards Topics: Immigration and Islam

We also conduct a sentiment analysis to investigate whether, beyond talking more about Islam and immigration, German users also develop more negative attitudes toward these topics. An increase in the negativity would indicate that we do not observe a simple national tendency towards talking more about core AfD messagebut rather a shift in opinions towards AfD attitudes.

We manually classify a random sample of 2000 users' tweets about Islam and immigration in three classes: negative, neutral, and positive. We train different classifiers with the `sklearn` library in Python and find that the best one is a Ridge classifier with 9 cross-fold validation for model evaluation. This classifier reaches an accuracy score of 73%. We then look at the coefficients of each feature (i.e., word) in order to understand the most indicative feature of each class. We identify the following words for negative tweets: kriminell (criminal), illegal_einwand (illegal immigration), kippah, islamist_gefahrd (islamist_danger), koran, migrant, illegal,stopislam, massenmigration (mass migration), islamisier, armutsmigration (poverty migration), islamist. Tweets containing these words always have negative sentiment.

We hence go back to our sample of not-classified tweets and classify tweets containing these words as having a negative sentiment. These are the words for neutral tweets: unterschied (difference), littlewiseh, kommentar (comment), muench (Munich), lauft (running), fussball, centrum, weltbank_unterwandert (world bank infiltrate), bergwand (mountain wall). These are the words for positive tweets: menschenrecht (human rights), europa, brauch zuwander (need immigrants), prophet, islam wert (Islam value), ramadan, rechtsstaat (rule of law), nichtohnemeinkopftuch (not without a headscarf), verabschiedet (adopted). We hence classify the non-classified tweets containing these words as having neutral/positive sentiment. In this way, we have a much bigger sample of classified tweets for our classifier model to learn. We hence re-train a new set of classifiers on this large sample and find that this time the best one is a Bernoulli classifier, which reaches an accuracy of 96.2%.

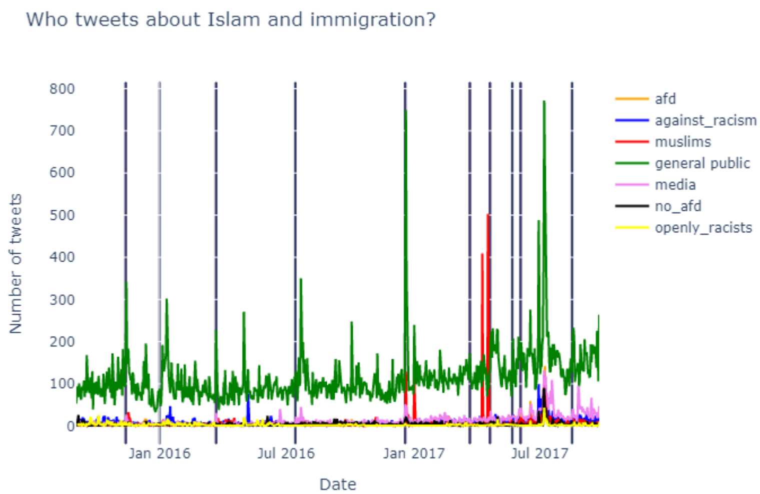
We then randomly sample close to 700 tweets and use a human coder to assess their sentiment. We evaluate the precision, recall, and F1-score of the predictions against this human coding. Precision is 0.84, recall 0.74, and F1-score 0.75. These findings corroborate our confidence in the generalizability of the classifier.

Who tweets about Islam and immigration?

After showing that, over time, German users tend to discuss more core AfD messages related to Islam and immigration, and with increasingly negative attitudes, we analzye who drives this conversation. We select those tweets that our LDA model classifies as discussing Islam or immigration with a confidence higher than 0.90. Next, we identify the top 5% users who post the highest number of these tweets. The tweets of these 231 “prolific” users represent 17% of the total Islam-immigration tweets. Then, we manually read the description in these users Twitter account as well as a few tweets and classify them in seven categories: AfD politicians (excluding the official party account), politicians by other parties (excluding the official party accounts), media, users who are openly racists, users who are clearly against racism, Muslims, and general public.

We report in Figure B.1 below the number of tweets that these categories post. The vertical bars represent the days of the events. We observe that the general public always posts the majority of the Islam-immigration tweets. Consistent with the LDA analysis, we observe a positive trend as well as that these tweets peak around the days of the event. We observe a peak of tweets by the general public on July 7th, 2017, consistent with the findings of the LDA analysis. Also, we observe a peak of tweets by Muslims on April 17, 2017. We manually read these tweets and found out that these tweets are about the fact that Mirza Masroor Ahmad, the leader of the worldwide Ahmadiyya Muslim Community, visited Germany.

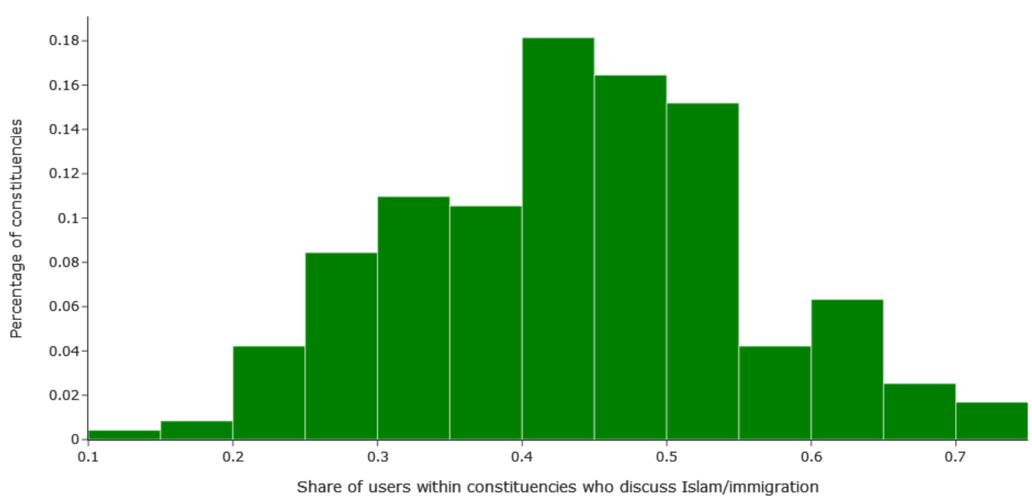
Figure B.1: Daily Tweets about Islam and Immigration by Most Prolific Users



Notes: Figure B.1 shows the number of tweets about Islam and immigration by category. The vertical bars represent the days of the events.

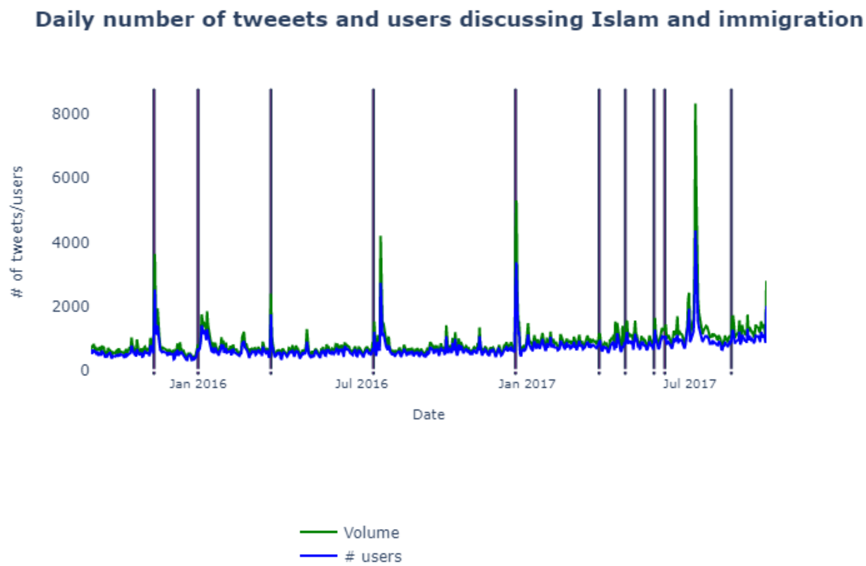
Furthermore, we identify, within each constituency, the Twitter users who discuss immigration and Islam. We find that the mean of users within a constituency discussing these topics is 0.43 and the standard deviation is 0.12, indicating that patterns in word similarity are due to a broad range of individuals and that there is large variance across constituencies. Figure B.2 a provides a visualization.

Figure B.2: Salience, Sentiment, and Users tweeting towards Immigration and Islam



(a) Variation in Salience

Notes: Subfigure B.2(a) visualized the percentage of Twitter users who discuss immigration and Islam within constituencies.



(b) Correlation between Users and Volume

Notes: Subfigure B.2(b) shows the daily number of Islam-immigration tweets and the daily number of users posting these tweets over time.

Finally, we provide evidence whether changes in the salience of the topics of consideration are due to more individuals tweeting on Islam and immigration, or just those individuals who tend to tweet about this topics tweeting more. We compute the daily number of Islam-immigration tweets and the daily number of users posting these tweets. We plot these two time series in Figure B.2b. We find an almost perfect correlation of 0.98 between the two time series, indicating that the increase in volume is due to more users start tweeting rather than to a few users tweeting more.

Summarizing, the compound of LDA, sentiment, and most prolific users analyses reveal five key findings. First, over time, the general public tends to discuss more two core AfD topics such as Islam and immigration. Second, this discussion seems to be natural and not driven by medias or politicians. Third, the discussion shows an upward trend and it peaks around the days of the events. Fourth, there is also an increasingly negative attitude when discussing these topics. Fifth, it is not just a small group of users increasing their Twitter output about the topics of consideration, but a general increase of users tweeting about these topics. We believe that these findings provide some explanations about the similarity trend toward AfD that we observe in our data.

Table B.1: LDA: Seeds for each Topic

migration
zuwander, fluchen, fluchtling, asyl, migration, polizei, einwander, asyl, asylrecht, asylunterkunft, asylrechtverscharf, asylbewerb, fluchtling, syri, pegid, verletzen, gewalt, angreifen
islam
islam, jihad, muslim, minarett, burka, scheiss, polit, hass, terror, angst, welt, gewalt, opf, anschlag, gesellschaft, medi, land, freiheit
elections
wahl, btw, polit, merkel, partei, deutschland, vote, schulz, bundestag, herr, wahlkampf, gewahlt, bundestagswahl, land, ergebnis
soccer
fussball, tor, fan, bundesliga, saison, punkt, spielen, vfb, platz, team, gewinnen, fan, bvb, borussia_dortmund, mannschaft
world politics
trump, turkei, deutschland, europa, erdogan, eu, usa, merkel, land,polit, welt,russland, brexit
education
schul, deutschland, stadt, fordern, zukunft, bildung, land, thema, ford, studi, polit, wirtschaft, arbeit, bildung
economy
euro,troik, geld, wirtschaft, arbeit, kauf, eur, zahl, wirk, steuer, steu
arts (music and film)
liv, ticket, film, konzert, feiern, party, musik, festival, band, album, cinema, kunst
digital
digital, digitalisi, zukunft, twitt, digitalisier, facebook, fb, googl, follow, youtub, retweet, stream, follow, onlin, blog
cities
dusseldorf, koln, berlin, stuttgart, leipzig, mainz, munch, frankfurt, bay, nuremberg, hanov
spare_time
urlaub, bier, leck, schlaf, kaffe, haus, trink, kuch, freund, wein, pizza,, vegan, fruhstuck, schmecken, rezept, heiss, koch
house
haus,geburstag,freund,feiern,family,katz,hund, haus, freund, bett, schlaf, sitzen, katz, bleib, arbeit, geburtstag, hund, weihnacht
mobility
auto, bahn, fahrt, fahr, zug, bus, flug, flucht, flugzeug, autobahn, richtung, sbahn, bahnhof
social networks
appl, updat, android, ios, test, microsoft, app, twitt, gewinn, retweet, facebook, stream, eur, tweet, youtub, cool, follow, liv, gewinnspiel, schauen, instagram, onlin
information
schreib, les,versteh,artikel, lern, artikel, versteh, fall, antwort, text, buch
interview
gest, thomas, buch, gluckwunsch, gast, guest, interview, gesprach, gluckwunsch

Notes: seed words for the topics classified by the Latent Dirichlet Allocation (LDA) model to classify the content of parties and public tweets.

C Appendix: Text Processing Details

As discussed in the main text, we compute similarity between the tweets of the parties and the tweets of each constituency by transforming the two groups of tweets into vectors using doc2vec, a deep learning model that we describe below. We then measure similarity as the cosine similarity between the two vectors. Before proceeding with doc2vec, we pre-process tweets. In the following paragraphs we provide details on these steps.

Text Preprocessing

Text pre-processing is necessary to reduce the computational time necessary to run the doc2vec model. Computational time is more than directly proportional to vocabulary size, namely the number of words in our corpus of tweets. With pre-processing we reduce the number of words, and hence computational time, without losing relevant information. We follow standard procedures in text pre-processing with different libraries in Python. First we lower-case all words and tokenize the text, i.e., we break streams of text into single words, called "tokens". We do this using "word_tokenize" from the Python module NLTK. Next, we eliminate punctuation and stop words, namely words that recur very frequently in our corpus and have little meaning. The dictionary of stop words we use is the one in NLTK. We also remove all tokens that consist of non-alphanumeric characters only, and remove emoticons, links, @, and # symbols. Then, we perform "stemming", which implies conflating the variant forms of a word into a common representation, the stem. For instance, the words "ate" and "eating" are both reduced to the common stem "eat". Stemming relies on existing dictionaries: we use the German Stemmer in the Python module `gensim`. Finally, we perform collocations, namely, we identify combinations of two words that have a higher probability of occurring together than separately. For instance, the tokens "angela" and "merkel" have higher chances of co-occuring as the bigram "angela merkel" than separately. In this case, collocations transform the two separate tokens into just one: "angela_merkel". We used `BigramCollocationFinder` in NLTK. We then use the pre-processed tweets to train the doc2vec model.

doc2vec

After pre-processing our tweets, we create two "documents", each at daily frequency: a party- and a constituency- document. The party document is the text of all the tweets the party posted on a certain day. A constituency document is the text of all the tweets that all the users in our sample located in a given constituency posted on a certain day. Since we have 752 days in our observation period (from September4th 2015 to September 24th, 2017,[4] we end up with 752 documents for each party and 752 documents for each constituency in our sample.[5]

We use doc2vec (Le and Mikolov 2014), an unsupervised deep learning algorithm that learns how to represent each document with a unique vector, and which is a generalization of Word2Vec. In order to understand doc2vec it is necessary to first understand how Word2Vec works. Word2Vec (Mikolov et al. 2013) is an unsupervised deep learning algorithm that learns how to represent each word as avector, depending on the surrounding (context) words. It takes as input a large vocabulary of words, trains a neural network language model with a single hidden layer, and produces a vector space, where each word is represented as a vector in this space. Word vectors, also called word embeddings, are positioned in the vector space such that words with similar semantic meaning are located in close proximity to one another. The model is trained using stochastic gradient descent with backpropagation. When the algorithm converges, it represents words as word embeddings, namely meaningful real-valued vectors of configurable dimension (usually, 300 dimensions).

doc2vec (Le and Mikolov 2014) is an extension of Word2Vec which learns to represent not just individual words, but entire documents. By treating each document as a word token, the same Word2Vec methodology is used to learn document embeddings (Bhatia et al. 2016). As in Word2Vec, training happens through backpropagation through several iterations. Each iteration of the algorithm is called an "epoch", and its purpose is to increase the quality of the output vectors. This type of document embedding allows to represent texts as dense fixed-length feature vectors, taking into account their semantic and syntactic structure.

We use the Distributed Bag of Words (DBOW) model and a freely available implementation of the doc2vec algorithm included in the `gensim` Python module, whose implementation requires the following hyperparameters:

• Size: the dimensionality of the vector representing the document. We set it to 300.

• Window size: The maximum distance between the current and predicted word within a sentence. We set it to 15.

• Epochs: Number of iterations over the corpus to train the algorithm. We set it to 300.

• Min_count: Ignores all words with total frequency lower than this. We set it to 20.

• Sub-sampling: The threshold for configuring which higher-frequency words are randomly down-sampled: useful range is (0, 10^{-5}). We set it to 10^{-3}.

• Negative: The number of "noise words" that should be drawn. We set it to 5. With the resulting measures, we compute the cosine similarity described in the main text.

---

[4] As stated in the main text, July 2015 marked a turning point in the history of AfD. We leave two months between the change in leadership of AfD and the starting point of our analysis, but we emphasize that the empirical method chosen is not sensitive to the exact day

[5] 752 is the maximum possible amount for a constituency whose users posted tweets every single day.

D Appendix: Validation

In order to control whether the computed similarity is indeed a valid measurement for how close public opinion is to the various parties, in this section we perform two validation experiments. These experiments first perform a basic but intuitive check by comparing the performance of doc2vec in assessing how close two documents of texts are, with the assessment by a human reader. Then, we focus on the German public and use public opinion and electoral data and observe that they also are correlated with our measure of similarity.

Human Reader

At the end of this section (Table D.2 – Table D.5) we present tweets of German parties and constituencies from four different days, together with the computed language similarity, which were assessed by a native German who first read both the tweets within a constituency and the corresponding party tweet on a given day and then judged between high or low similarity. The answers confirm the high and low text similarity computed by the doc2vec algorithm. This validation is only on a basic level, and is no proof of a valid similarity, but it presents a first transparent way to assess the quality of our measurement.

Votes and Polls

To further control the validity of the results of our doc2vec model, we compute the correlation between our measure of similarity and a) the results of the 2017 federal election at constituency level, and b) poll data provided by Infratest Dimap (2018) at state level.

For the election outcomes, we merge the tweets posted in the 30 days before the election within electoral constituencies and then apply the doc2vec algorithm. We repeat this 15 days before the election as a robustness check. The reason for merging texts over 30 days is to produce a sufficient amount of text for both parties and constituencies as not all parties posted tweets in the days immediately before the election. For the analysis of poll data, since poll surveys are conducted at state level, we merge the tweets of all the constituencies in a given state on the day of the poll.

We then perform two regression analyses: one with the change in vote share from2013 to 2017 for all parties as the dependent variable, and one with the poll results as the dependent variable.[6] In both cases, we regress the dependent variables on the measured similarity. We cluster standard errors on the lowest aggregate for the units of observation, i.e. electoral constituency level or state level, respectively. For the regression on poll results in levels we include party fixed effects to control for variations in levels of party support. Results are presented in Table D.1. We observe a

[6] Each observation is party-constituency (in case change in vote share is the electoral outcome as dependent variable) or party-state-date (in case the poll results at state level on a given day is the dependent variable).

positive correlation in all analyses. This analysis offers further support for the fact that our computed similarity captures the public mood across states and electoral borders.

Table D.1: Validation: Measured Similarity

	ΔVote Share		**Poll**
	(1)	(2)	(3)
Similarity:			
15 days before Election	0.0987		
	(0.0171)		
30 days before Election		0.125	
		(0.0189)	
2015 to 2017			0.0679
			(0.0218)
Observations	1079	1151	454
R^2	0.021	0.032	0.684

Notes: ΔVote Share refers to the difference in electoral results between 2017 and 2013. The independent variable, similarity, is the measured similarity in the specified period. We merge text 15 and 30 days before the election. For polls, it corresponds to the day the poll was conducted. In estimations of electoral outcomes (1) and (2) standard errors are clustered on constituency level. Poll refers to poll surveys at state level conducted between 2015 and 2017. In estimation of poll results (3) standard errors are clustered at state level and party fixed effects are included. Standard errors in parentheses.

Table D.2: Example Tweets

Party Tweet (The Greens)

Raus aus der #Kohle , rein in die #Erneuerbaren ! Die #Klimaziele erreichen wir nur ohne Kohle. #endegelaende pic.twitter.com/4EPzxY4biB

Constituency Tweets

’Wie ging die Woche für Dich aus? Hast Du Deine geplanten Vorhaben verwirklicht? Was motiviert Dich....und was... https://t.co/Sl5Ku6mwkJ’
’24+24 kg Oberkörper Finisher https://t.co/3dZG8P3qiz’
’RT @dg_4und20: Chuck Norris ruft für sein eigenes Land an. #ESC2016 #esc #esc16 #eurovision’
’Österreich und Georgien nach vorne!! #aut #geo #ESC #esc2016 #eurovision
’Das erste Lied das sich ”anders” anhört und mit fast dezenter Bühnenshow , Franz Ferdinand meets Moby... like. #geo #Esc2016 #esc’
’So, Italien bringt was fürs Auge... Ach die singt auch? Gar nicht mitbekommen. #esc’
’Überschwemmung im Teletubbie Land #italy #esc’
’Ich habe bei dem Song immer das Gefühl jemand müsste von 33 auf 45umin umschalten #slothofgermany #ger #esc’
’Das erste Lied das sich ”anders” anhört und mit fast dezenter Bühnenshow, Franz Ferdinand meets Moby... like. #geo#Esc2016
@z3ktus er ist doch Longboard-Rider, da gehört das zum Style oder es hat ihn eben einfach hingefetzt #Lat #esc’
’Oh, Harald Glööökler macht jetzt Bühnenshows ... Naja Hauptsache es lenkt nicht vom eigentlichen Liedbeitrag ab #rus #esc’
’Ich höre nur ”let it go, let it goho”... #frozen #cro #esc
’Jetzt singt Morticia Adams schon für Serbia #srb #esc #Eurovision’
’Ich so: ”Jaaa, endlich Rocker !!” ...und dann spielen die Jungs so BonJovi meets David Guetta #schade #cyp #Eurovision’
’Das ist der Franzose. nein ! DOCH !! Oh!!! #fra #esc’
’Bei dieser Frisur weiß man wenigsten woher der Wind wehte #ltu #esc
’Ah ja. Für Leute die man auffordern muss die Folie zu entfernen BEVOR man die TK-Pizza in den Ofen schiebt :D https://t.co/lnpTeYardK
’Klar doch, meine sind alle aus Vollgummi :D https://t.co/FOYVJmTjOJ
’Schöne Pfingsten ;) https://t.co/skomZukepg’
’PISA! Noch Fragen? Kinder sind doch aus Vollgummi :D https://t.co/yolVgyeK1C’

Computed Similarity: -0.22
Assess Similarity: Low

Notes: comparison of the computed similarity between a party and a constituency on a given day with human evaluation.

Table D.3: Example Tweets

Party Tweet (AfD)
#AfD #Pazderski: „NO!”-Tattoos-“ Wie dämlich geht’s eigentlich noch? https://www.alternativefuer.de/pazderski-no-tattoos-wie-daemlich-gehts-eigentlich-noch/-
Constituency Tweets
’Neue Version der #IMSWARE APP HelpDesk veröffentlicht - Apple iOS https://t.co/omGbOiM8tI’, ’hat einen Runtastic Lauf über 3,63 km in 41m 56s mit der #Runtastic PRO App absolviert: https://t.co/g4ajk5FlBH’, ’@hassanscorner willst du Schalke siegen sehen , musst du Mittwoch ins Stadion gehen ..... yes WE can.... Wird aber schwer’ ’So wird es auch am mittwoch sein @s04 @hassanscorner @Sky_Dirk https://t.co/VgihbV5cgK’
Computed Similarity: -0.19 **Assess Similarity: Low**
Notes: comparison of the computed similarity between a party and a constituency on a given day with human evaluation.

Table D.4: Example Tweets

Party Tweet (CSU)

Um 19.10 Uhr: @CSU -Chef Horst #Seehofer im ZDF-Sommerinterview! Einschalten lohnt sich! #CSUtvTipp #berlindirekt
pic.twitter.com/ZP4DqkMW7Xpic.twitter.com/ZP4DqkMW7X
#Seehofer: Amoklauf in München aber auch die Attentate in Würzburg & Ansbach haben sich in unsere Herzen eingebrannt
pic.twitter.com/zlZpgcpuDDpic.twitter.com/zlZpgcpuDD
#Seehofer auf der Trauerfeier: #Sicherheit ist das höchste Gut einer Demokratie, die oberste Pflicht des Staates.
#Seehofer auf der Trauerfeier: Menschen in unserem Land haben ein Recht darauf, dass wir entschlossen gegen jede Form von Gewalt vorgehen.

Constituency Tweets

’Die #SPD-Kandidaten haben alle ’nen recht deutlichen Schatten. #agh16”’
’Bevor ein Bayer Bundeskanzler wird gehört Bayern zu Österreich. #Seehofer #csu’
’Flüchtlingskrise: Seehofer distanziert sich scharf von Merkels ”Wir schaffen das” https://t.co/SE4HRDiOTw’
’die französischen muslime verweigern den muslimischen attentätern die letzte ehre... ...gut so... ein klares... https://t.co/BqZTYsLVFZ’
’Eine Muslimin betete heute bei der Trauerfeier in München: ”Allah, beschütze diese schöne Stadt und ihre Bewohner’
’Das #Sommerinterview mit Horst #Seehofer fand ich gut. Aber einen eigenen #Kanzlerkandidaten der #CSU kann ich mir nicht vorstellen.’
’@tagesschau: Seehofer: Ein bisschen zurückrudern, ein bisschen nachlegen https://t.co/vwW36uoYZK #CSU #Flüchtlinge’
’Horst Seehofer gegen Angela Merkel: Er kann einfach nicht anders’
’Es gibt Krawatten, mit denen würde ich mir nach einem Unfall nicht mal das Bein abbinden’
’Der Münchner Attentäter war lt. @NDRinfo ”kein Rechtsextremist”. Halt bloß ein ganz gewöhnlicher Hitler-Fan.’
’Wir müssten uns mal erlauben,in der Türkei eine deutsche Demo abzuhalten. Es ist einfach unfassbar was wir hier alles gestatten. #koeln3107’
’Ich versuche mal zu schlafen. Kann grade nichts tun und nur hoffen, dass die Einheimischen gestern Recht hatten.#Kreta #Waldbrand’
’Warum überträgt der Ereigniskanal Phoenix nicht die Trauerfeier in München? Schwer zu verstehen!’
’Wer für Erdogan auf d Straße geht unterstützt #Terror gegenüber Andersdenkenden #ARDSommerinterview’
’Wer hat Gauck und Merkel zur Trauerfeier eingeladen? Ist diese Inszenierung nicht erbärmlich?’
’Woher haben die Leute eigentlich immer Holzlatten und Eisenstangen? Flüchtlinge verprügeln Security-Mitarbeiter’
’@welt Wo wohnen bitte all diese Menschen?Was bietet die AfD denn für Lösungen?Und der München-Attentäter war AfD-Fan. https://t.co/rq9HfqKXhH’

Computed Similarity: 0.17
Assess Similarity: High

Notes: comparison of the computed similarity between a party and a constituency on a given day with human evaluation.

Table D.5: Example Tweets

Party Tweet (The Greens)
’Kommt alle am Wochenende zu den Menschenketten gegen #Rassismus ! http://www.gruene.de/menschenkettehttp://www. gruene.de/menschenkette #HandInHand pic.twitter.com/h4oSMNXs9ipic.twitter.com/h4oSMNXs9i’
Constituency Tweets
’Bahnhöfe: Licht am Ende der Brücke aus Berliner Abendblatt: Licht am Ende der BrückeNach langem Hin und Her w... https://t.co/yPf71k2PaL’
’@Schmidtlepp Barcodes, der Kitt von Weltreichen.’
’Bus + Straßenverkehr: Den meisten Falschparkern fehlt das Unrechtsbewusstsein… https://t.co/728xdipm32’
’@spdberlin danke liebe SPD für eine Regenwanderschaft. Veranstaltung so gut organisiert wie die Bildungspolitik!’
’@UlrichSchulte Dont́ mess with ”polizeilichen Befugnissen”! https://t.co/MVxQYaMpaO’
’#EnthemmteMitte zeigt: Antisemitismus,Antiziganismus, Homophobie; Verschwörungstheorien sind Probleme aller Parteien https://t.co/0yCLplvj55’
’Es gibt übrigens immer noch keine Neuigkeiten von Frank. Danke für euren Support. https://t.co/jneDmOmZL3’
’Die gute Nachricht: #EheFürAlle finden inzwischen alle gut. Die schlechte Nachricht: #KüssenVerboten #EnthemmteMitte https://t.co/Rt6pbwJfWE’
’#HandinHand gegen Rassismus! Komm zur Menschenkette in Berlin’
’#HandinHand gegen Rassismus! Komm zur Menschenkette in Bochum, Berlin, Hamburg, München und Leipzig https://t.co/qMkcOyFOOi via @campact’
’HTW_Berlin: @miauzus @rbb24 Die HTW Berlin wird den Lehrauftrag von Wolfgang Hebold sofort beenden.Wir dulden weder Rassismus noch Fremdenfeindlichkeit’
’Das neue Berlin wird aus Europaletten und Überseecontainern errichtet.’
’Potsdam: Öffentlicher Nahverkehr in Potsdam Der Herr der Schienen, aus \xa0PNN https://t.co/HYUcXsd3jD’
’Russland führt Hooligan-EM-Tabelle an, England auf Platz 2, Deutschland nur dritter #hooligans #em2016’
’S-Bahn: Hohe Hürden für S-Bahn bis Rangsdorf, aus\xa0MAZ https://t.co/my2m0XtueO’
’Schön, dass jemand den Rassismus erkennt: https://t.co/2p2J0XFnwE’
’Schutzblechen’
’@Tagesspiegel @spdberlin und Staatssekretär für Bildung war glatt entgangen, dass man in Schulen nicht wahlkämpft... https://t.co/CwAhBlcZ4j’
’U-Bahn Wegen Graffiti-Schäden fahren Züge mit weniger Waggons’
’Es ist übrigens auch #Rassismus, mich als Rassistin zu beschimpfen, nur weil ich blond bin und blaue Augen habe.’
Computed Similarity: 0.55
Assess Similarity: High

Notes: comparison of the computed similarity between a party and a constituency on a given day with human evaluation.

E Appendix: Discontinuous Growth Model

Variables considered include:

1. *Time*: The first variable represents the linear time trend found in a typical growth model.

2. *Time*2: Similar to before with a quadratic time trend.

3. *E*: Event specific change in intercept variable coded 0 prior to the event and1 after the event, until the next event occurs.

4. *Reset*: Event specific change in slope variable coded 0 at the period of which the event first occurs and increases with each subsequent period until the nextevent.

5. *Reset*^{2}: Similar to before with a quadratic change variable.

For analyzing multiple events, we simply introduce multiple variables for events and changes. The following table offers an overview on the coding of variables:

Table E.1: Coding of Time Variables - Multiple Events

Time	$Time^2$	E_1	E_2	$Reset_1$	$Reset_1{}^2$	$Reset_2$	$Reset_2{}^2$
0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
2	4	0	0	0	0	0	0
3	9	0	0	0	0	0	0
4	16	0	0	0	0	0	0
5	25	1	0	0	0	0	0
6	36	1	0	1	1	0	0
7	49	1	0	2	4	0	0
8	64	1	0	3	9	0	0
9	81	1	0	4	16	0	0
10	100	0	1	0	0	0	0
11	121	0	1	0	0	1	1
12	144	0	1	0	0	2	4
13	169	0	1	0	0	3	9
14	196	0	1	0	0	4	16

Notes: example of the coding of time variables for the discontinuous growth model in case of two events. The first event occurs in period 5, the second event in period 10.

F Appendix: Results

The following tables complement the choice of the functional form of the discontinuous growth model (DGM) introduced in equation 1 in the main text, as well as results visualized and discussed in the main text.

Table F.1 shows the results of likelihood ratio tests for the null hypothesis of having a better fit by only including a linear time trend into the discontinuous growth models for each party, compared to the alternative of adding also a quadratic term (no event variables included). As one can see, the null hypothesis is rejected in all cases at the 0.1 percent significance level, and for most cases even at the 0.01 percent significance level. We thus include the quadratic time trend in the DGM for all parties.

Table F.2 shows the full list of estimated coefficients for the DGM for each party.

Table F.3 shows the estimated coefficients for the within-party discontinuous growth models for AfD and SPD, visualized in Figure 4b and Figure 4c in the main text.

Table F.4 shows an investigation at the potential determinants of the heterogeneity in estimated random coefficients of the main DGM. The explanatory variables used corresponds to a set used by Franz et al. (2018) to explain the electoral success of the AfD after the 2017 general election. Notice that the optimal set of explanatory variables may vary across parties, but for the sake of comparison we used the same explanatory variables in each regression.[7] As discussed in the main text, we do not find any significant relationship of these variables with the magnitude of the estimated random coefficients.

Table F.5 shows the results of estimating the discontinuous growth model on a set of sport events, namely:

- the *DFB Pokal Finale* (German soccer league final) on May 21, 2016
- the match Germany vs. Italy in the UEFA Euro league 2016 on July 02, 2016
- the match Germany vs. France in the UEFA Euro league 2016 on July 07,2016
- the *DFB Pokal Finale* (German soccer league final) on May 27, 2017

[7] Notice that the only variation in the set of explanatory variables is caused by the CSU not existing in the eastern part of Germany, hence the East indicator is excluded.

Table F.1: Likelihood Ratio Test Results: Linear vs Quadratic Time Trend

Party	Likelihood Ratio Test Statistic	p-Value
AfD	804.49	<0.01
CDU	498.4	<0.01
CSU	3.53	0.06
FDP	35.13	<0.01
SPD	56.46	<0.01
The Greens	238.91	<0.01
The Left	3.8	0.05

Notes: Test results refer to a likelihood ratio test for the null hypothesis of a better fit using only a linear time trend versus the alternative of a quadratic term.

Table F.2: Discontinuous Growth Model Results

	AfD	CDU	CSU	FDP	The Greens	The Left	SPD
Time	0.000162	-0.000167	-0.000045	-0.000728	0.000635	-0.000111	-0.001485
	(0.000199)	(0.000202)	(0.000413)	(0.000167)	(0.000180)	(0.000176)	(0.000172)
Time^2	-0.000017	0.000004	-0.000004	0.000009	-0.000005	-0.000004	0.000018
	(0.000003)	(0.000003)	(0.000006)	(0.000002)	(0.000003)	(0.000002)	(0.000002)
Event$_1$	-0.189245	-0.129344	0.015701	0.021558	-0.020901	-0.015322	0.011962
	(0.004852)	(0.005022)	(0.010167)	(0.004345)	(0.004446)	(0.004474)	(0.004337)
Event$_2$	0.200228	-0.036637	0.057938	-0.071373	-0.003188	0.063699	-0.096552
	(0.020061)	(0.020331)	(0.041784)	(0.016965)	(0.018157)	(0.017830)	(0.017397)
Event$_3$	0.544867	-0.112406	0.159906	-0.226041	0.057046	0.184897	-0.444325
	(0.075773)	(0.076876)	(0.157436)	(0.063891)	(0.068505)	(0.067287)	(0.065530)
Event$_4$	1.562768	-0.340869	0.378014	-0.691941	0.216601	0.405414	-1.346470
	(0.216918)	(0.220021)	(0.450713)	(0.182824)	(0.196064)	(0.192559)	(0.187527)
Event$_5$	3.710215	-0.847889	0.878455	-1.653615	0.758808	0.866092	-3.354696
	(0.532342)	(0.539945)	(1.106049)	(0.448660)	(0.481144)	(0.472546)	(0.460194)
Event$_6$	5.344926	-1.286950	1.309918	-2.423111	1.185146	1.267024	-4.923720
	(0.783015)	(0.794199)	(1.626826)	(0.659919)	(0.707705)	(0.695055)	(0.676887)
Event$_7$	5.913527	-1.457644	1.418320	-2.713813	1.403728	1.395900	-5.517458
	(0.871035)	(0.883477)	(1.809691)	(0.734102)	(0.787261)	(0.773187)	(0.752976)
Event$_8$	6.554457	-1.572828	1.607238	-3.016718	1.594557	1.525893	-6.213154
	(0.973610)	(0.987517)	(2.022825)	(0.820552)	(0.879971)	(0.864241)	(0.841650)
Event$_9$	6.841765	-1.647187	1.637697	-3.158336	1.652775	1.586773	-6.426340
	(1.013526)	(1.028003)	(2.105733)	(0.854192)	(0.916047)	(0.899671)	(0.876155)
Event$_{10}$	8.525083	-2.076569	2.043185	-3.993125	2.100466	1.942572	-8.143523
	(1.277501)	(1.295749)	(2.654166)	(1.076668)	(1.154633)	(1.133992)	(1.104351)
Event$_{11}$	9.276724	-2.273887	2.205123	-4.376092	2.282414	2.087538	-8.912369
	(1.393212)	(1.413113)	(2.894580)	(1.174189)	(1.259216)	(1.236705)	(1.204379)
Reset$_1$	0.005987	0.007958	-0.000962	-0.002376	0.001379	0.005159	-0.001407
	(0.000399)	(0.000403)	(0.000822)	(0.000340)	(0.000363)	(0.000357)	(0.000348)
Reset$_2$	0.003424	-0.001545	-0.000230	-0.001467	-0.001857	0.002149	-0.002578
	(0.000501)	(0.000508)	(0.001040)	(0.000423)	(0.000453)	(0.000445)	(0.000433)
Reset$_3$	0.003903	-0.001965	0.000731	-0.003270	0.000060	0.001800	-0.005859
	(0.000928)	(0.000942)	(0.001929)	(0.000783)	(0.000839)	(0.000824)	(0.000803)
Reset$_4$	0.010319	-0.002964	0.002050	-0.004997	0.003663	0.002237	-0.009772
	(0.001559)	(0.001581)	(0.003239)	(0.001314)	(0.001409)	(0.001384)	(0.001348)
Reset$_5$	0.015100	-0.005738	0.003973	-0.007197	0.006585	0.004192	-0.015143
	(0.002446)	(0.002481)	(0.005082)	(0.002062)	(0.002211)	(0.002171)	(0.002115)
Reset$_6$	0.009798	-0.001697	-0.001753	-0.011161	0.013461	-0.000711	-0.022276
	(0.003046)	(0.003090)	(0.006325)	(0.002568)	(0.002754)	(0.002705)	(0.002634)
Reset$_7$	0.017491	0.005055	0.004213	-0.010487	0.005747	-0.002687	-0.019454
	(0.003181)	(0.003227)	(0.006608)	(0.002682)	(0.002875)	(0.002824)	(0.002750)
Reset$_8$	0.020368	-0.011068	-0.010721	-0.023622	0.003075	0.001506	0.010854
	(0.004246)	(0.004309)	(0.008804)	(0.003578)	(0.003837)	(0.003769)	(0.003670)
Reset$_9$	0.020834	-0.006731	0.004702	-0.010329	0.004244	0.004030	-0.022386
	(0.003367)	(0.003415)	(0.006994)	(0.002837)	(0.003043)	(0.002988)	(0.002910)
Reset$_{10}$	0.025353	-0.007087	0.003990	-0.013292	0.004995	0.004525	-0.026293
	(0.003839)	(0.003894)	(0.007981)	(0.003237)	(0.003470)	(0.003409)	(0.003319)
Reset$_{11}$	0.016587	-0.020151	-0.001945	-0.038246	0.006985	0.007723	-0.021844
	(0.005272)	(0.005352)	(0.010917)	(0.004443)	(0.004765)	(0.004680)	(0.004557)
Reset$^2_1$	0.000072	-0.000228	0.000047	0.000026	-0.000013	-0.000076	-0.000024
	(0.000007)	(0.000007)	(0.000015)	(0.000006)	(0.000007)	(0.000007)	(0.000006)
Reset$^2_2$	0.000015	0.000012	0.000015	-0.000005	0.000034	-0.000014	-0.000019
	(0.000003)	(0.000003)	(0.000007)	(0.000003)	(0.000003)	(0.000003)	(0.000003)
Reset$^2_3$	0.000054	-0.000001	0.000011	-0.000005	0.000016	-0.000003	-0.000017
	(0.000003)	(0.000003)	(0.000006)	(0.000002)	(0.000003)	(0.000003)	(0.000003)
Reset$^2_4$	0.000018	-0.000003	0.000007	-0.000006	-0.000002	0.000004	-0.000018

	(0.000003)	(0.000003)	(0.000006)	(0.000002)	(0.000003)	(0.000002)	(0.000002)
Reset^2_5	0.000025	0.000013	0.000004	-0.000014	-0.000015	-0.000003	-0.000025
	(0.000003)	(0.000003)	(0.000007)	(0.000003)	(0.000003)	(0.000003)	(0.000003)
Reset^2_6	0.000407	-0.000224	0.000229	0.000047	-0.000229	0.000161	0.000061
	(0.000026)	(0.000026)	(0.000053)	(0.000022)	(0.000023)	(0.000023)	(0.000022)
Reset^2_7	0.000095	-0.000452	-0.000004	0.000014	0.000003	0.000228	-0.000074
	(0.000020)	(0.000020)	(0.000041)	(0.000017)	(0.000018)	(0.000018)	(0.000017)
Reset^2_8	0.000340	0.000600	0.001190	0.001004	0.000181	0.000086	-0.004121
	(0.000235)	(0.000238)	(0.000484)	(0.000198)	(0.000212)	(0.000208)	(0.000203)
Reset^2_9	0.000026	0.000018	0.000008	-0.000019	0.000029	0.000011	-0.000011
	(0.000004)	(0.000004)	(0.000008)	(0.000003)	(0.000003)	(0.000003)	(0.000003)
Reset^2_{10}	-0.000037	0.000005	0.000025	0.000016	0.000082	0.000008	0.000026
	(0.000024)	(0.000024)	(0.000049)	(0.000020)	(0.000021)	(0.000021)	(0.000020)
Reset^2_{11}	0.001113	0.001619	0.000146	0.002983	-0.000228	-0.000078	-0.000315
	(0.000375)	(0.000381)	(0.000769)	(0.000316)	(0.000339)	(0.000333)	(0.000324)
2016 (Ind.)	0.018115	-0.009527	0.021905	0.010483	0.039278	-0.000681	-0.003011
	(0.004464)	(0.004531)	(0.009240)	(0.003763)	(0.004035)	(0.003963)	(0.003859)
Constant	0.339869	0.308465	0.342375	0.341589	0.313398	0.315637	0.341546
	(0.004271)	(0.004681)	(0.009660)	(0.004398)	(0.004401)	(0.004248)	(0.004301)

Notes: Maximum likelihood estimation results for the discontinuous growth models for all parties, corresponding to the visualizations in Figure 4a and 5 in the main text. All estimates refer to the fixed component (see Equation 1). Standard errors in parentheses.

Table F.3: Within-Party Discontinuous Growth Model Results

	AfD	SPD
Time	0.000394	0.139222
	(0.110865)	(0.076391)
$Time^2$	-0.00045	-0.01601
	(0.012025)	(0.008286)
$Event_1$	-0.0418	0.207223
	(0.267538)	(0.176741)
$Event_2$	0.068183	2.055247
	(0.705468)	(1.074187)
$Event_3$	0.175864	8.742369
	(1.579449)	(4.55106)
$Event_4$	0.169005	25.00546
	(3.654221)	(12.90163)
$Event_5$	0.739519	62.67071
	(15.98357)	(32.33963)
$Event_6$	1.202165	94.8064
	(27.09273)	(48.46349)
$Event_7$	1.487453	104.013
	(31.8545)	(53.66692)
$Event_8$	2.632928	117.2648
	(38.34794)	(60.54409)
$Event_9$	2.708277	120.3558
	(39.71864)	(61.9692)
$Event_{10}$	3.346502	152.4504
	(56.38522)	(78.73957)
$Event_{11}$	3.683225	165.4034
	(63.16712)	(85.33497)
$Reset_1$	0.002155	0.165403
	(0.412545)	(0.105785)
$Reset_2$	-0.10885	0.376404
	(0.304352)	(0.195334)
$Reset_3$	-0.06041	0.76106
	(0.307778)	(0.390408)
$Reset_4$	0.023697	1.277179
	(0.422298)	(0.654913)
$Reset_5$	0.055648	1.998228
	(0.883692)	(1.036623)
$Reset_6$	0.126646	1.547356
	(1.168377)	(1.278924)
$Reset_7$	0.822974	2.496242
	(1.25093)	(1.33947)
$Reset_9$	-0.00511	2.652491
	(1.384181)	(1.434036)
$Reset_{10}$	-0.01545	3.226702
	(1.665557)	(1.624527)
$Reset_{11}$	0.012843	3.245857

	(1.743795)	(1.682095)
$Reset^2_1$	0.042154	0.013956
	(0.191268)	(0.014351)
$Reset^2_2$	0.03935	0.014764
	(0.078853)	(0.008921)
$Reset^2_3$	0.009621	0.016737
	(0.028201)	(0.008407)
$Reset^2_4$	0.000381	0.015636
	(0.012099)	(0.008305)
$Reset^2_5$	-0.00127	0.017165
	(0.014512)	(0.008683)
$Reset^2_6$	-0.02732	0.227745
	(0.078853)	(0.054333)
$Reset^2_7$	-0.14323	0.039659
	(0.043357)	(0.029875)
$Reset^2_9$	0.006804	0.026164
	(0.01315)	(0.009061)
$Reset^2_{10}$	0.028884	-0.01424
	(0.078853)	(0.054333)
$Reset^2_{11}$	0.001229	0.01615
	(0.012026)	(0.008286)
2016 (Ind.)	-0.00134	0.011858
	(0.250863)	(0.110249)
Constant	0.009378	-0.14256
	(0.217449)	(0.149832)

Notes: Maximum likelihood estimation results for the within party discontinuous growth models for AfD and SPD corresponding to the visualizations in Figure 4b and 4c in the main text. All estimates refer to the fixed component (see Equation 1). The trend variable for event 8 is not included as the following event is one week apart and thus the trend cannot be identified with weekly time variable. Standard errors in parentheses.

Table F.4: Investigating Heterogeneity in Similarity Shifts

	AfD	CDU	CSU	The Greens	FDP	The Left	SPD
Age (60+)	0.0022957	51.1606313	-125490.7253961	-8766.6568494	72743.6774321	-9792.7161485	-15354.5355238
	(0.0127525)	(6913.2422613)	(524886.9793106)	(12901.6096456)	(39218.5328311)	(29747.1725311)	(28961.2281063)
Foreign Population (%)	0.0129157	5872.6108693	79573.5274925	15391.6823620	67223.0277461	37045.5233375	44778.4581600
	(0.0070703)	(3406.4478637)	(199560.2573615)	(7456.3460440)	(21370.7723677)	(15943.2817773)	(16136.5140401)
Disp. Income	-0.0000001	-3.7343483	-164.9006288	-5.8987860	-6.1199989	-36.7268055	-13.8832325
	(0.0000161)	(9.0732860)	(498.3000761)	(20.4908466)	(60.1709172)	(40.2522309)	(43.0512894)
Craftsmen Firms	-0.0138153	-7356.8551257	337815.2598775	7415.7479018	-24275.4203766	-29598.6707528	-42342.8301019
	(0.0270794)	(13926.6436485)	(436417.9420512)	(25107.7449804)	(90616.3773608)	(59163.4929417)	(68858.2983667)
Unemployment 2017	-0.0051535	-637.3449091	12107.2466249	-11065.8062762	-35300.5907563	14116.4337207	9391.8044638
	(0.0166029)	(8230.8861223)	(1226768.6538083)	(16491.7361461)	(59644.7404555)	(36743.0550428)	(40363.6485876)
High Education	-0.0000129	949.9003180	7377.4993253	7746.5580848	13813.0123785	4851.0191945	6434.4285624
	(0.0045261)	(2374.6790491)	(136508.5401287)	(4864.9470446)	(14888.6130221)	(10303.6982557)	(11096.6276003)
Manufacturing	-0.0016238	-962.9838709	-116062.0879633	-5324.1681884	-15097.0021293	-3174.6900533	-7866.4460467
	(0.0028057)	(1403.7832457)	(80012.6851666)	(3178.6364585)	(10664.4229868)	(7643.7638872)	(7390.2236982)
East	0.1294102	48554.1577101		209490.1275657	342580.4326380	380964.7738273	557562.8612416
	(0.1032769)	(53654.6691369)		(113317.6475989)	(351795.9768875)	(260082.4942150)	(269342.7487112)
Observations	235	199	36	235	235	235	235
R^2	0.028	0.052	0.119	0.104	0.080	0.082	0.112

Notes: Dependent variable is the shift in similarity to the specified parties after the last event. To improve readability, dependent variables are re-scaled by factor 10^8. Craftsmen Firms are computed as the amount per 1000 inhabitants. Standard errors in parentheses calculated using bootstrapping.

Table F.5: Discontinuous Growth Model: Sport Events Results

	AfD	CDU	CSU	FDP	SPD	The Greens	The Left
Time	-0.00109299	-0.00024009	-0.00003017	-0.00001322	-0.00003148	-0.00007141	0.00018055
	(0.00001735)	(0.00001624)	(0.00001465)	(0.00001503)	(0.00001510)	(0.00001569)	(0.00001545)
$Time^2$	0.00000130	0.00000046	-0.00000001	0.00000004	0.00000003	0.00000003	-0.00000024
	(0.00000003)	(0.00000003)	(0.00000002)	(0.00000002)	(0.00000003)	(0.00000003)	(0.00000003)
$Event_1$	0.11412411	0.04290170	0.01236712	0.02097155	-0.00136163	0.02528588	-0.01228253
	(0.00459503)	(0.00420001)	(0.00386641)	(0.00401562)	(0.00395419)	(0.00415956)	(0.00399120)
$Event_2$	0.14609481	0.04621330	0.00461153	-0.00153713	-0.20381707	0.00709182	-0.05378382
	(0.00860598)	(0.00820386)	(0.00721285)	(0.00745994)	(0.00763849)	(0.00790238)	(0.00785762)
$Event_3$	0.07639631	0.00917228	0.00758684	-0.02514460	-0.01089685	-0.00586923	-0.02974085
	(0.00296233)	(0.00303621)	(0.00319083)	(0.00307298)	(0.00321217)	(0.00319051)	(0.00313019)
$Event_4$	0.14580175	-0.01709263	0.01776192	-0.01658928	-0.01630931	0.01898061	0.00841205
	(0.00650709)	(0.00617692)	(0.00590024)	(0.00596182)	(0.00589090)	(0.00599740)	(0.00608145)
$Reset_1$	-0.00046094	-0.00277214	-0.00093517	-0.00213474	0.00006406	-0.00520722	-0.00094135
	(0.00044938)	(0.00041927)	(0.00036382)	(0.00037436)	(0.00038254)	(0.00039851)	(0.00039217)
$Reset_2$	-0.00994331	-0.09316505	-0.00186426	0.00846266	0.19485989	0.00318851	-0.02150679
	(0.00991183)	(0.00918251)	(0.00791841)	(0.00817086)	(0.00835700)	(0.00873663)	(0.00858608)
$Reset_3$	0.00041633	-0.00001589	-0.00003661	0.00023916	0.00012688	0.00010387	0.00012870
	(0.00002558)	(0.00002409)	(0.00002258)	(0.00002276)	(0.00002309)	(0.00002369)	(0.00002329)
$Reset_4$	-0.00080838	0.00001458	0.00002020	0.00048768	0.00053531	-0.00005005	-0.00004751
	(0.00009888)	(0.00009221)	(0.00008168)	(0.00008441)	(0.00008536)	(0.00008841)	(0.00008771)
$Reset^2{}_1$	0.00002137	0.00006718	0.00002621	0.00003746	-0.00000399	0.00013270	0.00003988
	(0.00001045)	(0.00000968)	(0.00000835)	(0.00000861)	(0.00000881)	(0.00000921)	(0.00000905)
$Reset^2{}_2$	0.00210620	0.02232141	0.00008960	-0.00095048	-0.03600088	-0.00026573	0.00981162
	(0.00237618)	(0.00220134)	(0.00189829)	(0.00195881)	(0.00200344)	(0.00209445)	(0.00205748)
$Reset^2{}_3$	-0.00000054	-0.00000021	0.00000020	-0.00000061	-0.00000034	-0.00000002	-0.00000022
	(0.00000007)	(0.00000006)	(0.00000005)	(0.00000006)	(0.00000006)	(0.00000006)	(0.00000006)
$Reset^2{}_4$	0.00000090	-0.00000512	-0.00000053	-0.00000670	-0.00000635	0.00000062	-0.00000023
	(0.00000076)	(0.00000071)	(0.00000061)	(0.00000063)	(0.00000064)	(0.00000067)	(0.00000066)
2016 (Ind.)	0.06356423	0.04217572	0.00238353	-0.00561187	0.00391214	-0.00234029	-0.00648141
	(0.00148501)	(0.00137715)	(0.00118996)	(0.00122760)	(0.00125514)	(0.00131160)	(0.00128867)
Constant	0.34439464	0.29375516	0.32473689	0.33315647	0.32419949	0.33104793	0.30430820
	(0.00349739)	(0.00330935)	(0.00359221)	(0.00369874)	(0.00359160)	(0.00363571)	(0.00353777)

Notes: Maximum likelihood estimation results for the discontinuous growth models for all parties. All estimates refer to the fixed component (see Equation 1). Standard errors in parentheses.

F.1 Appendix: Additional results

F.1.1 Alternative models

In this section we estimate models that account for possible autocorrelation in the daily constituency-to-party similarity. Table F.6 reports estimates from a DGM with a one-day lag of the dependent variable. This specification is demanding and for some parties the model doesn't converge. Table F.7 instead reports estimates from dynamic panel models with constituency fixed effects, lag of the dependent variable and autocorrelated error term.

Table F.6: Discontinuous Growth Models with lagged dependent variables

	AfD	CSU	FDP	SPD
$Simil_{t-1}$	0.345059	0.162798	0.193988	0.147630
	(0.002267)	(0.005997)	(0.002366)	(0.002381)
Time	0.000345	-0.000102	-0.000632	-0.001439
	(0.000198)	(0.000429)	(0.000174)	(0.000179)
$Time^2$	-0.000016	-0.000002	0.000008	0.000017
	(0.000003)	(0.000006)	(0.000002)	(0.000002)
$Event_1$	-0.118348	0.010843	0.019015	0.010792
	(0.004501)	(0.009930)	(0.004070)	(0.004171)
$Event_2$	0.170330	0.040560	-0.060307	-0.094220
	(0.019385)	(0.042253)	(0.017057)	(0.017617)
$Event_3$	0.490900	0.105403	-0.192189	-0.428294
	(0.074036)	(0.160764)	(0.065023)	(0.067111)
$Event_4$	1.403924	0.237045	-0.587627	-1.295008
	(0.212478)	(0.461390)	(0.186571)	(0.192565)
$Event_5$	3.345991	0.544193	-1.408950	-3.228246
	(0.522115)	(1.133702)	(0.458446)	(0.473169)
$Event_6$	4.836273	0.817581	-2.065295	-4.741536
	(0.768277)	(1.668178)	(0.674585)	(0.696251)
$Event_7$	5.359231	0.875208	-2.313634	-5.309794
	(0.854723)	(1.855871)	(0.750489)	(0.774594)
$Event_8$	5.946404	0.994979	-2.570140	-5.985310
	(0.955467)	(2.074633)	(0.838947)	(0.865893)
$Event_9$	6.206942	1.009452	-2.693007	-6.179919
	(0.994673)	(2.159743)	(0.873372)	(0.901426)
$Event_{10}$	7.752670	1.251170	-3.406129	-7.837394
	(1.253958)	(2.722726)	(1.101038)	(1.136405)
$Event_{11}$	8.439869	1.353859	-3.726730	-8.573637
	(1.367621)	(2.969523)	(1.200841)	(1.239414)
$Reset_1$	0.004757	-0.000780	-0.002088	-0.001455
	(0.000377)	(0.000817)	(0.000334)	(0.000345)
$Reset_2$	0.003014	-0.000381	-0.001211	-0.002487
	(0.000487)	(0.001058)	(0.000428)	(0.000442)
$Reset_3$	0.004328	0.000278	-0.002749	-0.005619
	(0.000909)	(0.001974)	(0.000799)	(0.000824)
$Reset_4$	0.009426	0.001170	-0.004244	-0.009412
	(0.001529)	(0.003321)	(0.001343)	(0.001386)
$Reset_5$	0.014002	0.002470	-0.006134	-0.014599
	(0.002401)	(0.005213)	(0.002108)	(0.002176)
$Reset_6$	0.011849	-0.002784	-0.009533	-0.021214
	(0.002985)	(0.006479)	(0.002622)	(0.002706)
$Reset_7$	0.016625	0.002345	-0.008865	-0.018992

	(0.003120)	(0.006773)	(0.002740)	(0.002827)
$Reset_8$	0.020944	-0.009321	-0.021559	0.014035
	(0.004113)	(0.008905)	(0.003612)	(0.003728)
$Reset_9$	0.019278	0.002665	-0.008833	-0.021812
	(0.003306)	(0.007179)	(0.002903)	(0.002996)
$Reset_{10}$	0.022894	0.002444	-0.011060	-0.025116
	(0.003767)	(0.008180)	(0.003308)	(0.003414)
$Reset_{11}$	0.017084	-0.005845	-0.035181	-0.022730
	(0.005100)	(0.011018)	(0.004478)	(0.004621)
$Reset^2_1$	0.000043	0.000037	0.000023	-0.000019
	(0.000007)	(0.000015)	(0.000006)	(0.000006)
$Reset^2_2$	0.000016	0.000012	-0.000005	-0.000018
	(0.000003)	(0.000007)	(0.000003)	(0.000003)
$Reset^2_3$	0.000039	0.000008	-0.000005	-0.000017
	(0.000003)	(0.000006)	(0.000002)	(0.000003)
$Reset^2_4$	0.000017	0.000005	-0.000006	-0.000017
	(0.000003)	(0.000006)	(0.000002)	(0.000003)
$Reset^2_5$	0.000021	0.000002	-0.000012	-0.000023
	(0.000003)	(0.000007)	(0.000003)	(0.000003)
$Reset^2_6$	0.000258	0.000201	0.000041	0.000055
	(0.000024)	(0.000052)	(0.000021)	(0.000022)
$Reset^2_7$	0.000072	0.000000	0.000009	-0.000058
	(0.000019)	(0.000041)	(0.000017)	(0.000017)
$Reset^2_8$	0.000084	0.000924	0.001021	-0.004341
	(0.000222)	(0.000479)	(0.000195)	(0.000202)
$Reset^2_9$	0.000022	0.000007	-0.000016	-0.000007
	(0.000004)	(0.000008)	(0.000003)	(0.000003)
$Reset^2_{10}$	-0.000019	0.000005	0.000003	0.000021
	(0.000022)	(0.000048)	(0.000020)	(0.000020)
$Reset^2_{11}$	0.000769	0.000370	0.002886	-0.000090
	(0.000355)	(0.000762)	(0.000312)	(0.000322)
2016 (Ind.)	0.012285	0.019273	0.008852	-0.001808
	(0.004233)	(0.009157)	(0.003717)	(0.003836)
Constant	0.220686	0.287374	0.276170	0.294169
	(0.003638)	(0.009164)	(0.003960)	(0.004054)

Notes: Maximum likelihood estimation results for the discontinuous growth models with 1-day lag of similarity. All estimates refer to the fixed component (see Equation 1). Standard errors in parentheses.

Table F.7: Dynamic panel models with AR(1) disturbance

	AfD	CDU	CSU	SPD	FDP	The Left	The Greens
$Simil_{t-1}$	0.591852	0.435641	0.463051	0.447844	0.488626	0.429973	0.443978
	(0.001953)	(0.002374)	(0.005403)	(0.002155)	(0.002107)	(0.002179)	(0.002164)
Time	0.000521	0.000175	-0.000332	-0.000914	-0.000408	0.000206	0.000677
	(0.000158)	(0.000179)	(0.000361)	(0.000150)	(0.000145)	(0.000158)	(0.000161)
$Time^2$	-0.000015	-0.000000153	0.000002	0.000011	0.000005	-0.000006	-0.000007
	(0.000002)	(0.000002)	(0.000005)	(0.000002)	(0.000002)	(0.000002)	(0.000002)
$Event_1$	-0.070910	-0.078302	0.002954	0.008957	0.013189	0.002182	-0.011237
	(0.003565)	(0.004007)	(0.008055)	(0.003349)	(0.003234)	(0.003525)	(0.003580)
$Event_2$	0.146132	-0.010056	0.001298	-0.061561	-0.038872	0.060733	0.010227
	(0.015264)	(0.017280)	(0.034788)	(0.014460)	(0.013960)	(0.015249)	(0.015493)
$Event_3$	0.454895	-0.009934	-0.024745	-0.274895	-0.124076	0.212272	0.116726
	(0.058580)	(0.066351)	(0.133556)	(0.055509)	(0.053585)	(0.058548)	(0.059494)
$Event_4$	1.294443	-0.015795	-0.114473	-0.827137	-0.378831	0.549506	0.392600
	(0.168350)	(0.190718)	(0.383841)	(0.159539)	(0.154006)	(0.168285)	(0.171010)
$Event_5$	3.104656	-0.021297	-0.304462	-2.058805	-0.907918	1.271952	1.117841
	(0.413984)	(0.469030)	(0.943910)	(0.392332)	(0.378721)	(0.413857)	(0.420568)
$Event_6$	4.503757	-0.044190	-0.436033	-3.021539	-1.331419	1.873685	1.716330
	(0.609307)	(0.690342)	(1.389268)	(0.577444)	(0.557410)	(0.609134)	(0.619015)
$Event_7$	4.995692	-0.054857	-0.509429	-3.376106	-1.492072	2.073730	1.948342
	(0.677903)	(0.768067)	(1.545675)	(0.642458)	(0.620166)	(0.677716)	(0.688711)
$Event_8$	5.552375	0.022844	-0.558615	-3.827981	-1.654870	2.292776	2.191628
	(0.757849)	(0.858652)	(1.727962)	(0.718224)	(0.693304)	(0.757643)	(0.769937)
$Event_9$	5.795342	-0.038025	-0.598335	-3.930041	-1.735934	2.391572	2.276883
	(0.788960)	(0.893902)	(1.798896)	(0.747711)	(0.721766)	(0.788747)	(0.801546)
$Event_{10}$	7.257753	-0.040915	-0.775951	-4.998677	-2.197546	2.977540	2.885527
	(0.994725)	(1.127050)	(2.268064)	(0.942722)	(0.910009)	(0.994467)	(1.010607)
$Event_{11}$	7.911499	-0.041399	-0.838121	-5.455813	-2.398132	3.243054	3.135526
	(1.084931)	(1.229260)	(2.473739)	(1.028216)	(0.992536)	(1.084652)	(1.102258)
$Reset_1$	0.004016	0.004639	-0.000598	-0.001176	-0.001455	0.002747	0.001074
	(0.000292)	(0.000331)	(0.000665)	(0.000277)	(0.000267)	(0.000292)	(0.000296)
$Reset_2$	0.002860	-0.000593	-0.000816	-0.001567	-0.000760	0.001917	-0.000430
	(0.000384)	(0.000435)	(0.000876)	(0.000364)	(0.000352)	(0.000384)	(0.000390)
$Reset_3$	0.004540	-0.000297	-0.000968	-0.003561	-0.001747	0.002450	0.001247
	(0.000720)	(0.000816)	(0.001642)	(0.000683)	(0.000659)	(0.000720)	(0.000732)
$Reset_4$	0.008862	-0.000315	-0.001139	-0.005984	-0.002710	0.003478	0.004151
	(0.001213)	(0.001374)	(0.002765)	(0.001149)	(0.001109)	(0.001212)	(0.001232)
$Reset_5$	0.013298	-0.001042	-0.001351	-0.009266	-0.003976	0.006097	0.007478
	(0.001905)	(0.002158)	(0.004343)	(0.001805)	(0.001742)	(0.001904)	(0.001935)
$Reset_6$	0.013269	0.001893	-0.005689	-0.013439	-0.006221	0.003368	0.010168
	(0.002368)	(0.002683)	(0.005397)	(0.002244)	(0.002167)	(0.002367)	(0.002406)
$Reset_7$	0.016487	0.005832	-0.002480	-0.013457	-0.005696	0.002683	0.007136
	(0.002475)	(0.002804)	(0.005642)	(0.002346)	(0.002264)	(0.002474)	(0.002514)
$Reset_8$	0.021277	-0.021096	-0.011052	0.017515	-0.015987	0.006985	0.005636
	(0.003323)	(0.003750)	(0.007539)	(0.003144)	(0.003035)	(0.003306)	(0.003358)
$Reset_9$	0.018369	-0.000703	-0.002539	-0.014397	-0.005761	0.007111	0.006991
	(0.002624)	(0.002973)	(0.005982)	(0.002486)	(0.002400)	(0.002623)	(0.002666)
$Reset_{10}$	0.021323	-0.000369	-0.002329	-0.015398	-0.006796	0.008481	0.007991
	(0.002989)	(0.003386)	(0.006812)	(0.002833)	(0.002734)	(0.002988)	(0.003036)
$Reset_{11}$	0.013069	-0.011792	-0.014357	-0.018756	-0.027184	0.004540	0.012303
	(0.004171)	(0.004695)	(0.009430)	(0.003941)	(0.003806)	(0.004139)	(0.004202)
$Reset^2_1$	0.000020	-0.000123	0.000018	-0.000007	0.000017	-0.000031	-0.000005
	(0.000006)	(0.000006)	(0.000013)	(0.000005)	(0.000005)	(0.000006)	(0.000006)
$Reset^2_2$	0.000014	0.000010	0.000004	-0.000012	-0.000003	-0.000004	0.000023

	(0.000003)	(0.000003)	(0.000006)	(0.000002)	(0.000002)	(0.000003)	(0.000003)
$Reset^2_3$	0.000029	0.000002	0.000002	-0.000011	-0.000003	0.000002	0.000013
	(0.000002)	(0.000003)	(0.000005)	(0.000002)	(0.000002)	(0.000002)	(0.000002)
$Reset^2_4$	0.000015	0.000001	-0.000001	-0.000011	-0.000004	0.000007	0.000003
	(0.000002)	(0.000002)	(0.000005)	(0.000002)	(0.000002)	(0.000002)	(0.000002)
$Reset^2_5$	0.000019	0.000010	-0.000003	-0.000015	-0.000007	0.000001	-0.000010
	(0.000003)	(0.000003)	(0.000006)	(0.000002)	(0.000002)	(0.000003)	(0.000003)
$Reset^2_6$	0.000162	-0.000133	0.000136	0.000026	0.000029	0.000114	-0.000090
	(0.000019)	(0.000022)	(0.000043)	(0.000018)	(0.000018)	(0.000019)	(0.000019)
$Reset^2_7$	0.000046	-0.000281	0.000004	0.000006	0.000006	0.000148	0.000014
	(0.000015)	(0.000017)	(0.000034)	(0.000014)	(0.000014)	(0.000015)	(0.000015)
$Reset^2_8$	-0.000068	0.001591	0.000701	-0.003734	0.000876	-0.000055	0.000169
	(0.000185)	(0.000208)	(0.000417)	(0.000175)	(0.000169)	(0.000183)	(0.000185)
$Reset^2_9$	0.000019	0.000012	0.000002	0.000002	-0.000009	0.000011	0.000019
	(0.000003)	(0.000003)	(0.000006)	(0.000003)	(0.000003)	(0.000003)	(0.000003)
$Reset^2_{10}$	-0.000008	-0.000000378	-0.000027	-0.000006	-0.000012	-0.000008	0.000049
	(0.000018)	(0.000020)	(0.000040)	(0.000017)	(0.000016)	(0.000018)	(0.000018)
$Reset^2_{11}$	0.001050	0.001465	0.000769	0.000391	0.002484	0.000511	-0.000344
	(0.000304)	(0.000339)	(0.000679)	(0.000286)	(0.000276)	(0.000298)	(0.000302)
2016 (Ind.)	0.007604	-0.004352	0.015426	-0.000330	0.005164	0.004182	0.030286
	(0.003329)	(0.003759)	(0.007549)	(0.003149)	(0.003041)	(0.003315)	(0.003367)
Constant	0.134564	0.169095	0.188446	0.190709	0.175870	0.176717	0.169131
	(0.003258)	(0.003503)	(0.007376)	(0.003070)	(0.002969)	(0.003105)	(0.003129)

Notes: Results from fixed-effects dynamic panel models with 1-day lag of similarity and auto-regressive disturbance. Standard errors in parentheses.

References

Scheuerman, Morgan Klaus, Wade, Kandrea, Lustig, Caitlin, and Brubaker, Jed R. (2020), How We've Taught Algorithms to See Identity: Constructing Race and Gender in Image Databases for Facial Analysis. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1), 1–35.

Wang, Zijian, Hale, Scott, Adelani, David Ifeoluwa, Grabowicz, Przemyslaw, Hartman, Timo, Flöck, Fabian, and Jurgens, David (2019), Demographic Inference and Representative Population Estimates from Multilingual Social Media Data. The World Wide Web Conference. WWW '19. San Francisco, CA, USA: Association for Computing Machinery, pp. 2056–2067