

Modeling Distractibility from Videos

Dominik Graetz¹

¹ University of Oregon

Author Note

This report was completed as part of the requirements of the Educational Data Science course sequence at the University of Oregon.

Correspondence concerning this article should be addressed to Dominik Graetz.
E-mail: dgrtz@uoregon.edu

Abstract

In every moment, humans can choose between completing a given task (exploitaion) or searching the environment for potentially more rewarding information or tasks (exploration). This paper conceputalizes the exploration of financially non-rewarding (but entertaining, or inherently interesting) stimuli in a situation in which a financially rewarding task is available as distraction. I aim to predict the time points at which participants are distracted. I am using data from an experiment in which subjects could work on a simple, rewarding task while distractors in the form of videos are shown. Distraction from the task caused by videos is measured using eye-tracking on a frame-by-frame basis. Based on a dataset with millions of rows, reduced to > 80 k observations, I trained Lasso Regression models, Ridge Regression models and Elastic Nets to predict distraction (gazing at videos) from experimental variables alone and from experimental variables and embeddings obtained for each frame in each video. Accuracy of these models is relatively high with > 80 %, however, investigation of confusion matrix measures reveals that this level of accuracy is largely due to the relatively low probability of distraction in the data. These models do not show satisfactory results on key metrics like the true positive rate. However, including video frame embeddings does increase the true positive rate significantly, implying that the content of a video does in part contribute to the distractive power of a video in this situation. Suggestions for model improvement are discussed.

Link to Github Repo: <https://github.com/dgraetz/VideoDistraction>

Link to Google Drive with all data: <https://drive.google.com/drive/folders/1jubBPuDnMHI2Z1b7GBoCmmmd2og4UGZF?usp=sharing>

Keywords: Machine Learning, Lasso Regression, Regularized Regression, Elastic Net, Distractibility, social media

Word count: 2872

Modeling Distractibility from Videos

Research Problem

In every moment, humans have the choice between two different actions: *exploitation*, in which they continue engaging in a rewarding task. Alternatively, when in the *exploration* mode, humans search the environment for potentially more rewarding, alternative things to do. The factors that determine disengagement from a rewarding task and exploratory behavior are unclear. Previous research suggests that switches from exploitation to exploration can be caused by increasing task uncertainty, elevated error rate in the current task, and lower task reward, or, together, task utility (for overviews, see Aston-Jones & Cohen, 2005; Cohen, McClure, & Yu, 2007). Here, I aim to predict human distractibility from short videos while they are working on a rewarding task. To achieve this, I conducted an experiment in which task utility (i. e., reward) is systematically manipulated between experimental blocks. Eye-tracking reveals overt shifts of attention - and this is the outcome I aim to predict from these two experimental variables: specifically, whether subjects look at the videos, or not. Additionally, I obtain frame-by-frame embeddings from ResNet-18 as predictors.

The general approach here is to compare the predictive performance of regularized regression models (i. e., lasso and ridge regression, and elastic net). It is of interest to me to which degree video content determines distractibility - thus, I will examine whether frame-level embeddings improve regularized regression performance by comparing models with and without embeddings.

This research is important because it can help further our understanding about the situations and the cognitive processes underlying elevated distractibility in populations with ADHD or older adults, given that they often appear more distracted. Moreover, it is crucial to understand the properties of stimuli that draw attention away from a target and thus, this research is basic research that is relevant for all applied scenarios in which

distractibility needs to be predicted - a goal for ergonomics, for instance to improve driver assistance systems.

Method

Data collection is still ongoing. Currently, data from 16 undergraduate students, collected at the University of Oregon, is available for the current analysis. However, for four subjects, eye tracking data is missing - therefore, models were trained using data from twelve subjects.

The experiment was programmed in MATLAB 2022a (The MathWorks Inc., 2022) using Psychtoolbox 3 (Brainard, 1997; Kleiner, Brainard, & Pelli, 2007; Pelli & Vision, 1997). In this experiment, participants perform a series of blocks of trials with a simple computer task. In each trial, an arrow is presented on the screen, pointing to directions of integer multiples of 90° (up, right, down, left). The arrow can appear in four different colors (red, green, blue, yellow). The subject completes a trial by pressing a key on the numpad dependent on the color (blue - 8 (up), green - 6 (right), red - 2 (down), yellow - 5 (left)). Each block is time limited to 60 seconds and participants can earn monetary incentives dependent on accurate performance in a block. Thus, the rational goal would be to “exploit” this task and perform fast and accurately to maximize the reward. However, participants could also “explore” the environment - in 50 % of the trials, videos appeared far or close to the left or right of the arrow. The pool of videos that was used for this experiment was generated from the social media website vine.co and consists of 500 videos with a duration of 3-6 seconds. For an example trial, see Figure 1. Using an SR Research Eyetracker, I measured the eye position on the screen at 1000 Hz.

Multiple experimental manipulations were implemented, but here, I will focus on the factors relevant for the current analysis. First, I included a reward manipulation - in half of the blocks, participants could earn/lose \$0.01 per correct/incorrect trial (high reward)

whereas in the other half, participants could only earn/lose \$0.001 per correct/incorrect trial (low reward condition). Participants' final reward was typically between \$4 and \$5.

Second, I manipulated the inter-trial-interval - in half of the blocks, the time between trials with a blank screen was set to 0.5 s, in the other half to 1.5 s. With this manipulation, I changed the possible exploitation rate per block, which has an impact on the rate of exploration, following from this formula:

$$timecost_{exploration} = \frac{time_{exploration}}{time_{exploitation}}$$

This implies that if a trial (including the inter-trial-interval) is long, the cost of exploration is relatively low, hence, a higher overall rate of checking the videos (and thus, a higher level of distraction) should be apparent. In other words, if a trial takes longer to complete, the relative time cost of exploration (assuming that the time needed to explore remains constant) is lower. In so far unpublished studies, our group confirmed that exploratory behavior does follow this prediction if the denominator is manipulated (for conference posters, see Ahmad, Grätz, & Mayr, 2023; Grätz, Fröber, & Mayr, 2022; Grätz & Mayr, 2023).

Behavioral results and checking behavior

Before I am presenting the models, it is helpful to visualize distractability in the current sample, dependent on the experimental conditions. For an initial analysis it was determined for each trial whether the subject inspected the video at least once (coded as 1 if yes, 0 otherwise). For each condition and subject, the average of this variable was then calculated, representing the average video checking probability. Statistical results will not be discussed here, but numerically, average effects are as predicted - longer inter-trial-interval and lower reward were expected to reduce the cost of exploring the videos and this trend seems to be present in the data (Figure 2). This figure also reveals

that video checking behavior shows large variation between individuals.

Data preprocessing

All analyses were conducted using R (R Core Team, 2023) in RStudio (Posit team, 2023). For data processing, wrangling, and visualization, the `rio` (Chan, Chan, Leeper, & Becker, 2021) and `tidyverse` (Wickham et al., 2019) packages were used. For parallel processing, functions from the packages `doParallel` (Microsoft & Weston, 2022a) and `foreach` (Microsoft & Weston, 2022b) were employed. Eye-tracking data files were imported using the `eyelinker` package (Barthelme, 2021).

To model the degree to which people are distracted in these situations, I used a more fine-grained approach with the goal to predict distractibility for each video frame.

Per experimental session, two data files are present - eye-tracking data contains information about gaze behavior during the experiment, the behavioral data file contains trial-level information about the experimental conditions, the video shown, and responses.

I first prepared the eye-tracking data. Each eye-tracking data file consists of a list of data frames. The raw eye-tracking data frame (for an example, see Table 2) consists of one row per millisecond with information about time stamp, and x and y position on the screen. From the message data frame within the eye-tracking data (for an example, see Table 3), I identified information about experimental blocks, trials and video frame onsets and merged this with the raw eye-tracking data described above. A separate eye-tracking data frame also provides information about blinks, most importantly blink onsets and ends. Time stamps during which blinks occurred were then removed from the enriched eye-tracking data frame. Now, for each millisecond, information about time (from trial onset), eye position, experimental blocks and trials and video frame number was present. I then merged the eye-tracking data with the behavioral data, specifically to add information about the video that was presented and its location (left/right, close/far from the task

stimulus). For each row, I then determined whether the gaze was inside the area the video was presented (coded as 1), or not (coded as 0) - this is my distraction indicator, and the outcome variable I aim to predict. This resulted in a data frame with highly redundant information because consecutive rows within a given frame could only differ with respect to the time stamp and the distraction indicator. I reduced the redundancy and the size of the data set by keeping only one row of per subject x trial x video frame. The distraction indicator is now coded as “check” if the gaze was inside the video ROI, and as “no_check” if the video was not inspected..

Obtaining video frame embeddings

Using `ffmpeg` and the `imager` (Barthelme, 2023) package in R, I saved each frame from each of the 500 videos as an image. Embeddings for each of these frames were then obtained using the `reticulate` package (Ushey, Allaire, & Tang, 2023) and the `img2vec` library (Safka et al., 2022). Resnet-18 was the model selected because its output vectors with a length of 512 are shortest relative to the other available models in the `img2vec` library.

The embeddings were then merged with the frame-by-frame data set above.

Description of the final data

The final data set consists of 87877.00 observations. Each observation represents a video frame presented in the experiment with the following (model-relevant) columns: ID, Time stamp, ITI condition, Reward condition, the 512 embeddings (all predictors) and the outcome, whether the gaze was on the video (coded as 1) or not (coded as 0). For example observations without embeddings, see Table 4. For mean and standard deviation information of the outcome per condition, refer to Table 1. Embeddings across all images and vector positions were on a range of [0.00; 13.78]

Description of the models

Models were built in R (R Core Team, 2023) using R Studio (Posit team, 2023) with the packages caret (Kuhn & Max, 2008), and glmnet (Tay, Narasimhan, & Hastie, 2023). Ridge regression, lasso regression and elastic net models were selected as predictive model candidates. All numeric predictors were standardized and factors were one-hot coded (subject ID, inter-trial-interval condition, reward condition). Models were trained on 90 % of the data with 10-fold cross-validation and evaluated on the remaining 10 % test set. For hyperparameter tuning, the respective parameters were initially selected to span a relatively wide possible range at which model fit is expected to be improved. Models were generated multiple times, each time reducing the range of the respective hyperparameters to the range at which model improvement was visible in the previous iteration to find the best hyperparameters. LogLoss was the metric by which the final model was selected for each procedure. Model performance on the test set will be evaluated based on overall accuracy, false positive, false negative, true positive and true negative rates.

Model Fit

The models were tested on 10 % data that was withheld from training the models. Accuracy, true positive, false positive, true negative and false negative rates were calculated for the entire test set, separate for regularized regression type (ridge regression vs. lasso regression vs. and elastic net) and predictors included (including frame embeddings vs. not including frame embeddings). Additionally, these metrics are also calculated within individuals so that model performance can be evaluated per subject.

Overall performance

Accuracy, true positive, true negative, false positive and false negative rates of all models are communicated in Figure 4. First of all, all models perform better when frame

embeddings are added as predictors. Accuracy for all models is above 0.50. Performance differences between the regression types are negligible for the accuracy metric. While this initially sounds like a good model performance, it is important to evaluate the other metrics. Importantly, given the relatively low rate of video inspection (see Figure 2 and Table 1), we must consider the other metrics. The true positive rate for all models is above 0.41. Relative to all other models, lasso regression with frame predictors identifies most of the video checks, with a true positive rate of 0.57. This rate is close to chance level which indicates that the models at hand perform poorly in this key metric. Performance of all models is good when it comes to identifying frames that were not checked (true negative rate), with a minimum of 0.50 across all models. Additionally, few frames were incorrectly classified as inspected, with a maximum false positive error rate of 0.50 across all models. Performance is again poor for the false negative classification - errors are lower than 0.59 for all models, and the lasso regression model with video frame predictors performs best with a false negative rate of 0.43.

The pattern of true negative and false positive metrics indicates a strong bias of the model to classify a frame as not being checked. This is clearly driven by the baseline differences in video checking rate (which is relatively low, see Figure 2). Since true positives and false negatives, as well as true negatives and false positives are not independent metrics, the symmetric pattern we observe is not surprising. While models with frame embeddings generally do perform better than those without, this difference is only meaningful for the true positive rate and the false negative rate - for these measures, ‘knowledge’ of image content significantly improves the prediction of video inspection by 23.00 %. It is also only for these metrics that performance differences by model become pronounced - lasso regression performs better than regardless of the availability of frame embeddings as predictors. However, even the best fitting model is still not very good at predicting frame-by-frame distraction. Additionally, it should be noted that the relatively high accuracy score cited above is misleading and driven by baseline differences in checking

behavior (i. e., participants did not check at a rate of 50 %, see Table 1).

Model performance on the subject-level

Model performance per individual for all models with and without frames as predictors is presented in Figure 5. There is a significant amount of between-subjects variability in how well the models can predict distractibility on a frame-by-frame basis. The range for accuracy is [0.54; 0.98], for the true positive rate [0.00; 0.77], for the true negative rate [0.62; 1.00], for the false positive rate [0.00; 0.38], and for the false negative rate [0.23; 1.00] across all models. Thus, for some subjects the model does relatively well, for some it does poorly. Interestingly, for nearly all subjects the model with frame embeddings predicts distractibility much better on all metrics.

Discussion & Conclusions

Overall, model performance is better when frame embeddings are included as predictors, specifically, they improve the true positive rate and reduce false negatives significantly. The best regularized regression model on nearly all metrics and even within subjects is lasso regression with the tuning hyperparameters of $\alpha = 1.0000$ and $\lambda = 0.0000$. Even without frames embeddings as predictors, lasso regression performs best, with the tuning hyperparameters of $\alpha = 1.0000$ and $\lambda = 0.0002$. However, the predictive power of these models is severely limited, with a high false negative and low true positive rates.

Overall, information about the frame content as measured with embeddings benefits true positive and false negative rates, however, it is questionable whether the improvement on these measures justifies the computational effort necessary to obtain frame-wise video embeddings.

Where does the low performance come from? There are individual differences between subjects regarding distractibility, the degree they react to differences in reward

sensitivity and inter-trial-intervals and personal interests. Given that one-hot coded dummy variables were included for each subject, overall distractibility is accounted for in the model. However, how people react to the experimental manipulations may depend on between-subjects variables, such as socio-economic status, the willingness to exert mental effort, and self-control, among other variables. Additionally, subjects have different interests and it seems reasonable to assume that subjects will be more distracted by videos that seem more interesting to them.

An alternative to the current models would be models that are trained individually for each subject or models that include the interaction between subject, experimental variables and video frame embeddings. This approach will be particularly helpful for subjects who deviate from the average behavioral pattern shown in Figure 2. This will be computationally much more intensive. Moreover, the audio component of the videos was not used as a predictor in the current models. Future models could obtain more holistic representation of entire video sequences or the audio component and add these representations as additional predictors.

Another problem with the current models is that they assume that the gaze reacts instantly to a frame on the screen. It is unclear what exactly the saccadic reaction time would be in this task, but it typically is in the order of 200 - 400 ms (Braun, Weber, Mergner, & Schulte-Mönting, 1992). The relationship between frame embeddings and gaze could be explored using cross-correlation and identifying the time lag at which these two variables correlate the strongest. For the models, the frame embedding predictors could be shifted by that time lag relative to the gaze data. As this increases the correlation between frame embeddings and gaze behavior, it should improve predictive model performance.

References

- Ahmad, S., Grätz, D., & Mayr, U. (2023). *Rational distraction: The adaptive nature of reliance on external action-relevant information*.
<https://doi.org/10.13140/RG.2.2.11482.93125>
- Aston-Jones, G., & Cohen, J. D. (2005). Adaptive gain and the role of the locus coeruleus–norepinephrine system in optimal performance. *Journal of Comparative Neurology*, 493(1), 99–110.
- Barthelme, S. (2021). *Eyelinker: Import ASC files from EyeLink eye trackers*. Retrieved from <https://CRAN.R-project.org/package=eyelinker>
- Barthelme, S. (2023). *Imager: Image processing library based on 'CImg'*. Retrieved from <https://CRAN.R-project.org/package=imager>
- Brainard, D. H. (1997). The psychophysics toolbox. *Spatial Vision*, 10(4), 433–436.
- Braun, D., Weber, H., Mergner, T., & Schulte-Mönting, J. (1992). Saccadic reaction times in patients with frontal and parietal lesions. *Brain*, 115(5), 1359–1386.
- Chan, C., Chan, G. C., Leeper, T. J., & Becker, J. (2021). *Rio: A swiss-army knife for data file i/o*.
- Cohen, J. D., McClure, S. M., & Yu, A. J. (2007). Should i stay or should i go? How the human brain manages the trade-off between exploitation and exploration. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 362(1481), 933–942.
- Grätz, D., Fröber, K., & Mayr, U. (2022). *Does distraction make you slow, or does being slow make you distracted? Testing a rational choice model of consulting the environment for action-relevant information*. <https://doi.org/10.13140/RG.2.2.12311.04004>
- Grätz, D., & Mayr, U. (2023). *Slower Response Speed Moves Us from Exploitation to Exploration*. <https://doi.org/10.13140/RG.2.2.36435.96802>
- Kleiner, M., Brainard, D., & Pelli, D. (2007). *What's new in psychtoolbox-3?*
- Kuhn, & Max. (2008). Building predictive models in r using the caret package. *Journal of Statistical Software*, 28(5), 1–26. <https://doi.org/10.18637/jss.v028.i05>

- Microsoft, & Weston, S. (2022a). *doParallel: Foreach parallel adaptor for the 'parallel' package*. Retrieved from <https://CRAN.R-project.org/package=doParallel>
- Microsoft, & Weston, S. (2022b). *Foreach: Provides foreach looping construct*. Retrieved from <https://CRAN.R-project.org/package=foreach>
- Pelli, D. G., & Vision, S. (1997). The VideoToolbox software for visual psychophysics: Transforming numbers into movies. *Spatial Vision*, 10, 437–442.
- Posit team. (2023). *RStudio: Integrated development environment for r*. Boston, MA: Posit Software, PBC. Retrieved from <http://www.posit.co/>
- R Core Team. (2023). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from <https://www.R-project.org/>
- Safka, C., yuiseki, Yedidi, K., Kozłowski, M., keherri, Bar, N., & Cobanov, M. (2022). Image 2 vec with PyTorch. <https://github.com/christiansafka/img2vec>; GitHub.
- Tay, J. K., Narasimhan, B., & Hastie, T. (2023). Elastic net regularization paths for all generalized linear models. *Journal of Statistical Software*, 106(1), 1–31. <https://doi.org/10.18637/jss.v106.i01>
- The MathWorks Inc. (2022). *MATLAB version: 9.12.0 (R2022a)*. Natick, Massachusetts, United States: The MathWorks Inc. Retrieved from <https://www.mathworks.com>
- Ushey, K., Allaire, J., & Tang, Y. (2023). *Reticulate: Interface to 'python'*. Retrieved from <https://CRAN.R-project.org/package=reticulate>
- Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., . . . Hiroaki Yutani. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1686. <https://doi.org/10.21105/joss.01686>

Table 1

Mean and standard deviation of the outcome variable in the final dataset, separate for each condition, computed on the frame-level.

ITI	RewardCond	M	SD
0.5	high	0.16	0.36
0.5	low	0.15	0.36
1.5	high	0.16	0.37
1.5	low	0.30	0.46

Table 2
*Example of 5 rows of the
raw data within the
eye-tracking file.*

time	xp	yp
5887302	986.00	606.00
5887303	985.10	606.50
5887304	984.20	606.60
5887305	983.10	606.40
5887306	983.00	606.20

Table 3

Example of 10 rows of the message data within the eye-tracking file.

time	text
6752096	BLOCK7_TRIAL9_FIXCROSS_ONSET
6752612	BLOCK7_TRIAL9_FIXCROSS_END
6752614	BLOCK7_TRIAL9_STIMONSET
6752616	BLOCK7_TRIAL9_VIDEOSTART
6752646	BLOCK7_TRIAL9_FRAME1
6752680	BLOCK7_TRIAL9_FRAME2
6752712	BLOCK7_TRIAL9_FRAME3
6752746	BLOCK7_TRIAL9_FRAME4
6752779	BLOCK7_TRIAL9_FRAME5
6752812	BLOCK7_TRIAL9_FRAME6
6752846	BLOCK7_TRIAL9_FRAME7

Table 4

The first ten rows of the training data set without frame embeddings. The training data set with frame embeddings is identical, but has 512 more columns in the format VX (where X stands for a number in the range [1; 512]) for the frame embeddings.

time	ID	ITI	RewardCond	isInVid
550	400	0.5	low	no_check
583	400	0.5	low	no_check
617	400	0.5	low	no_check
650	400	0.5	low	no_check
683	400	0.5	low	no_check
716	400	0.5	low	no_check
750	400	0.5	low	no_check
783	400	0.5	low	no_check
850	400	0.5	low	no_check
916	400	0.5	low	no_check

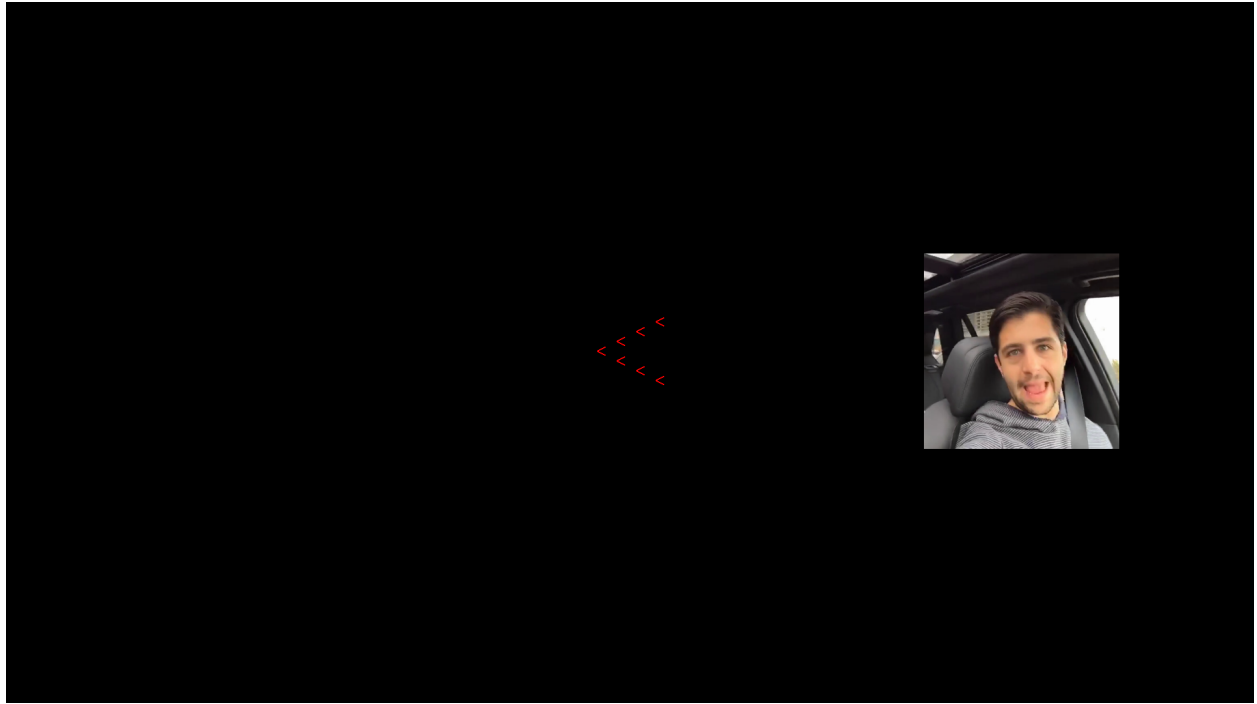


Figure 1. Example trial with a video on the far right to the central stimulus. The correct solution in the trial would be to press “2” because the stimulus is red.

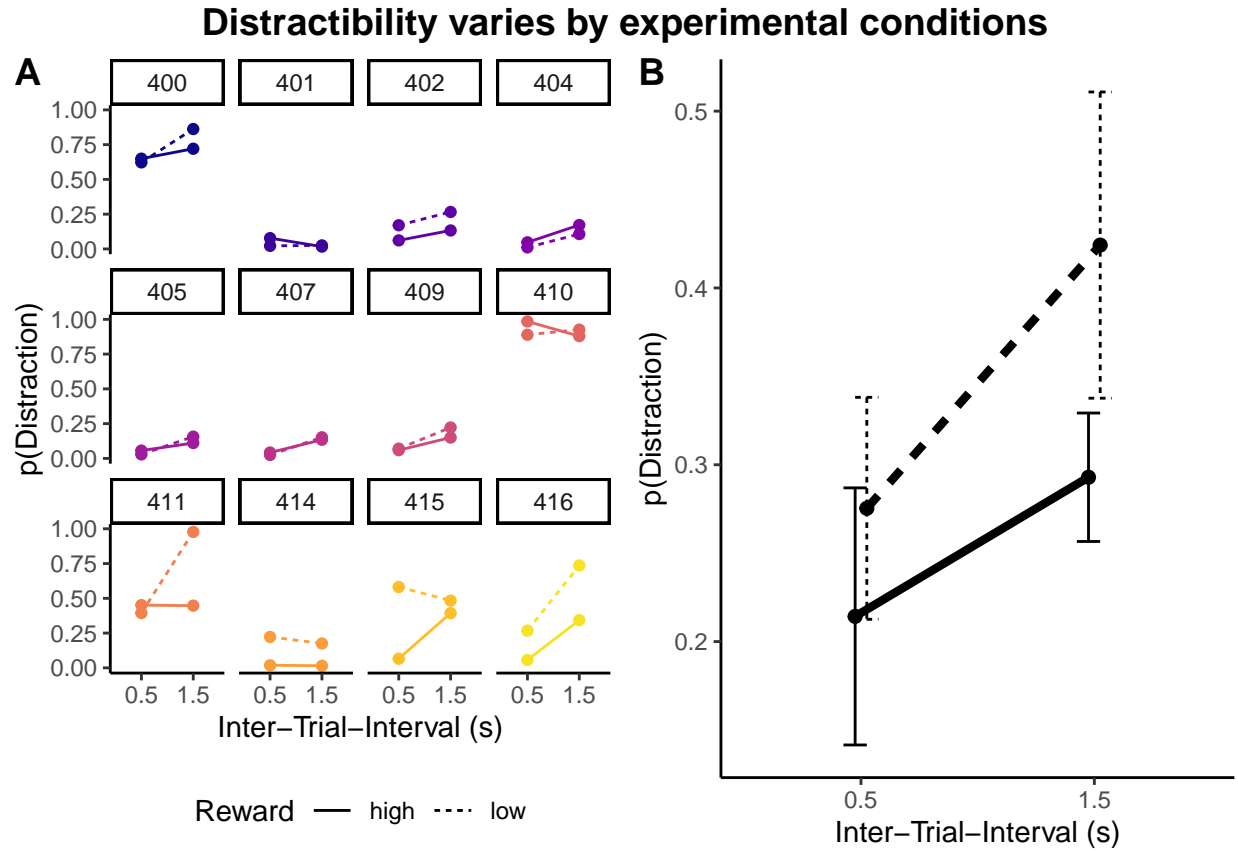


Figure 2. Behavioral results. Probabilities for distraction were obtained by dividing the number of trials with a gaze on the video by the number of trials, separately for each condition. **A:** Distractibility by subject and condition. **B:** Average effects. Errorbars represent within-subjects design corrected 95 % confidence intervals.

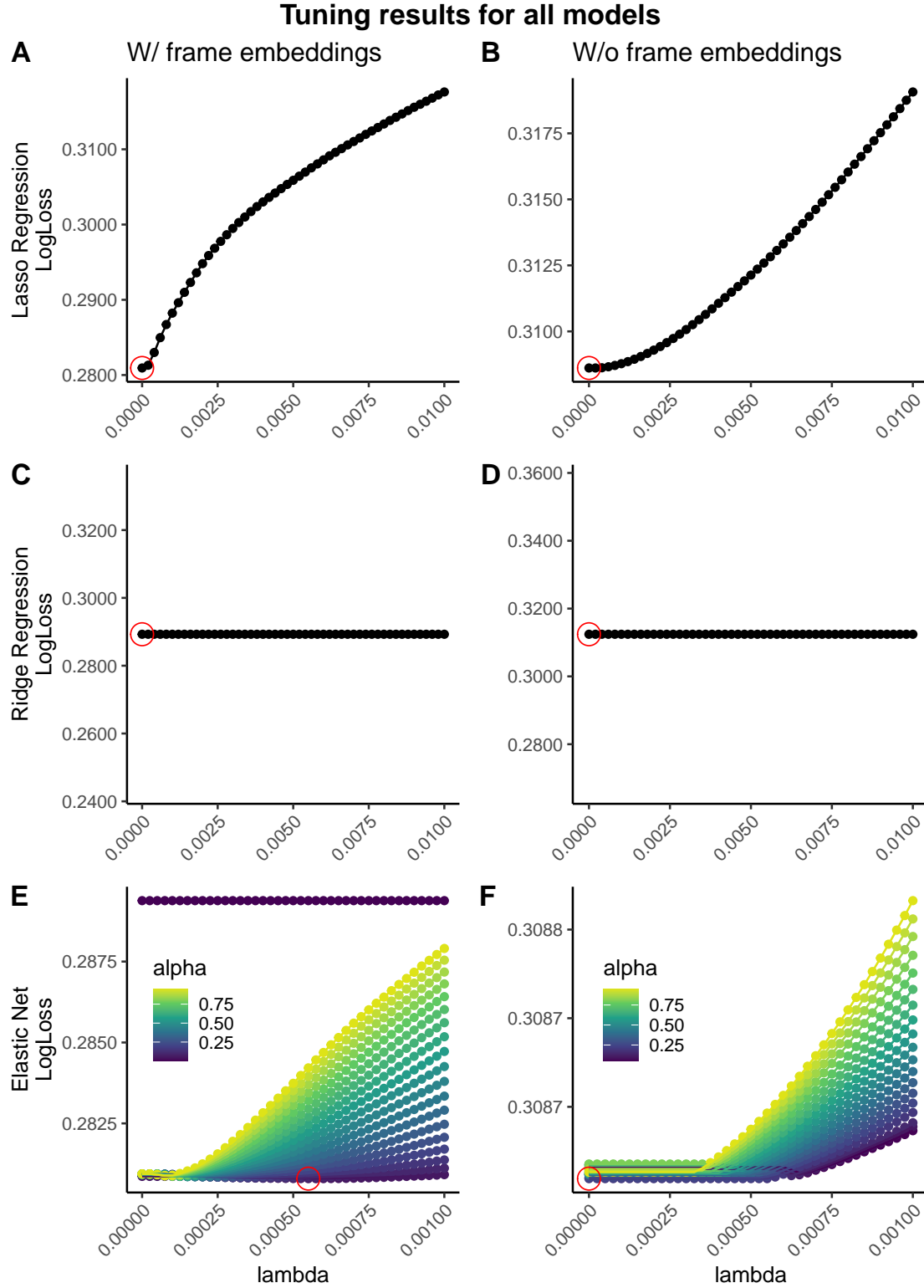


Figure 3. Tuning results for all models. The red circle represents the best tune out of the hyperparameter grid specified. Each row represent the tuning results for the regularized regression types (Lasso Regression: A, B; Ridge Regression: C, D; Elastic Net: E, F). Figures in the left column show the tuning results for the respective models including frame embeddings as predictors (A, C, E) or without frame embeddings as predictors (B, D, F).

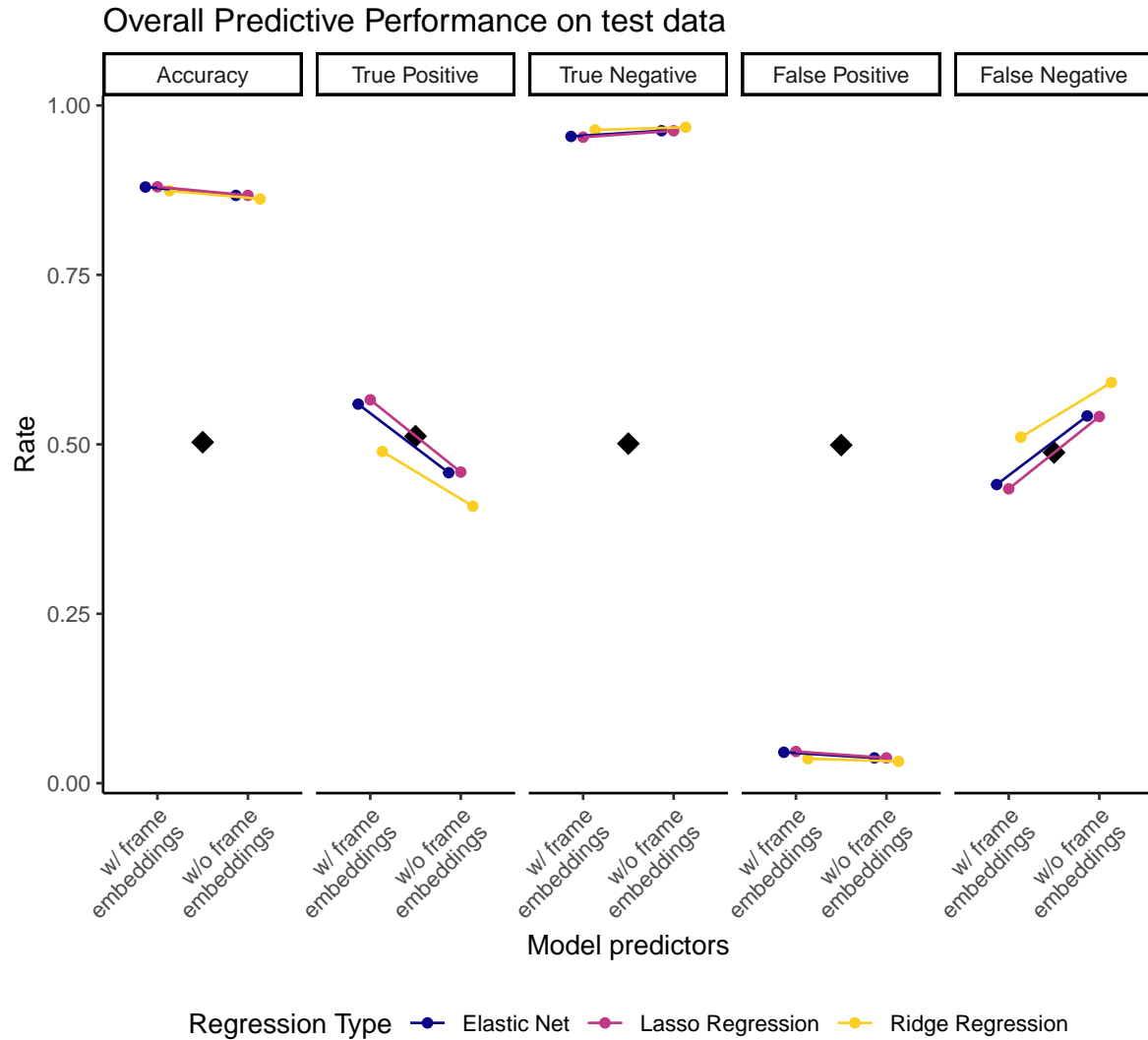


Figure 4. Performance of the regularized regression models with and without frame embeddings as predictors on the test set (10 % of the data). Black diamonds are chance level (from randomly assigning check vs. no check to each observation). Performance on all metrics is better if frame embeddings are available as predictors on all metrics. This effect is emphasized for the true positive and false negative rate. Performance differences between regularized regression type are negligible for accuracy, true negative and false positive rates and are more pronounced for true positive rates and false negative rates.

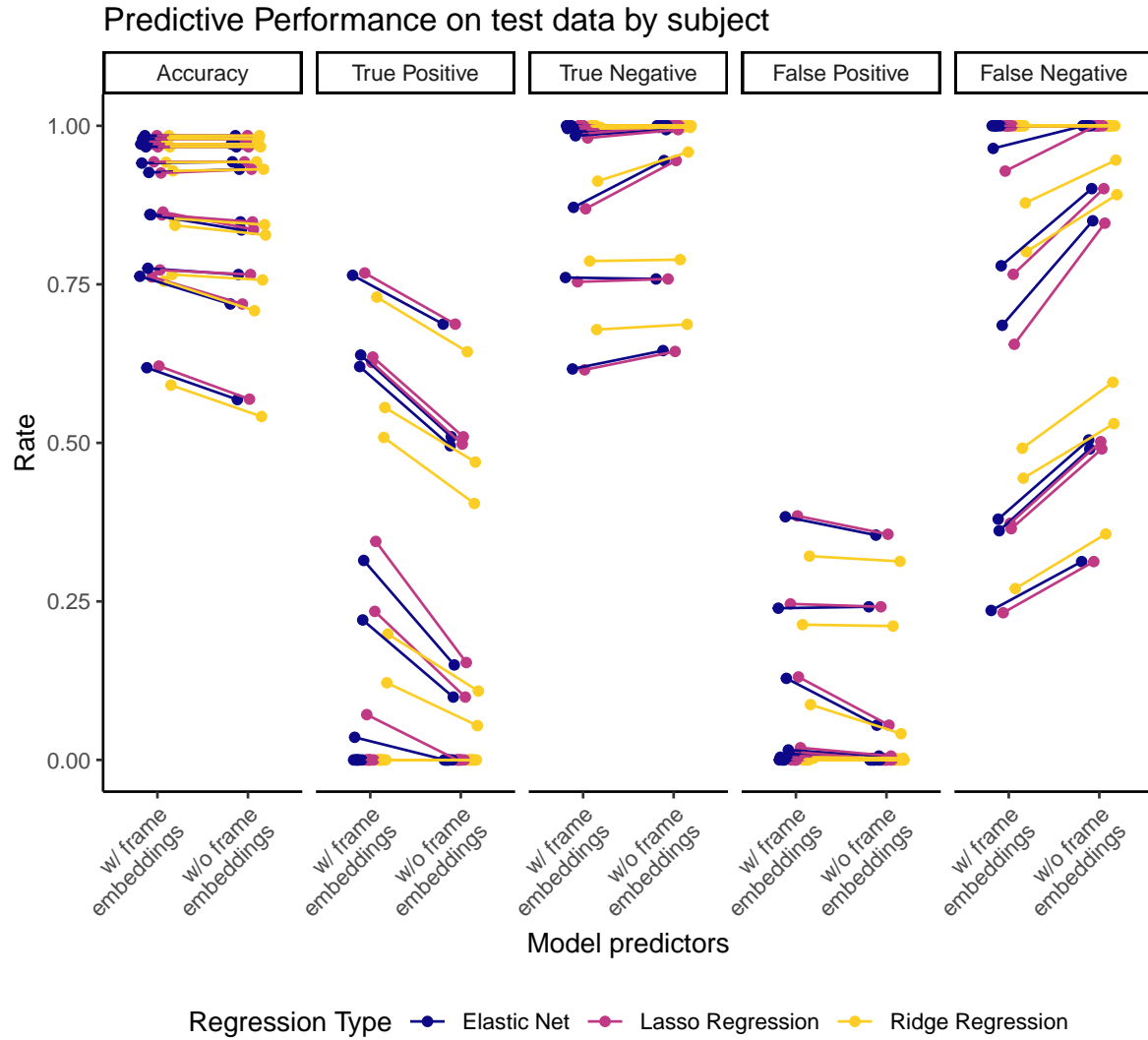


Figure 5. Performance of the regularized regression models with and without frame embeddings as predictors on the test set (10 % of the data), by subject. Note the differences in performance between subjects.