Arbitrary Experiment

```
library(conflicted)

library(kableExtra)
library(knitr)
library(broom.helpers)
library(broom)
library(dtplyr)
library(furrr)
```

```
## Loading required package: future
```

```
library(arrow)
library(glue)
library(fs)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
conflict_prefer("filter", "dplyr")
```

```
## [conflicted] Will prefer dplyr::filter over any other package.
```

```
source(here("analysis/utils.R"), local = knit_global())
set_theme()
```

```
write_bib(.packages(), here("analysis/packages.bib"))
sessionInfo()
```

```
## R version 4.4.0 (2024-04-24)
## Platform: aarch64-apple-darwin20
## Running under: macOS Sonoma 14.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: Asia/Singapore
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
```

```
##
## other attached packages:
##  [1] lubridate_1.9.3      forcats_1.0.0        stringr_1.5.1
##  [4] dplyr_1.1.4          purrr_1.0.2          readr_2.1.5
##  [7] tidyr_1.3.1          tibble_3.2.1         ggplot2_3.5.1
## [10] tidyverse_2.0.0      fs_1.6.4             glue_1.7.0
## [13] arrow_16.1.0         furrr_0.3.1          future_1.33.2
## [16] dtplyr_1.3.1         broom_1.0.6          broom.helpers_1.15.0
## [19] knitr_1.47           kableExtra_1.4.0     conflicted_1.2.0
## [22] here_1.0.1
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.5      xfun_0.45         tzdb_0.4.0        vctrs_0.6.5
##  [5] tools_4.4.0       generics_0.1.3    parallel_4.4.0    fansi_1.0.6
##  [9] pkgconfig_2.0.3   data.table_1.15.4 assertthat_0.2.1  lifecycle_1.0.4
## [13] compiler_4.4.0    munsell_0.5.1     codetools_0.2-20  htmltools_0.5.8.1
## [17] yaml_2.3.8        pillar_1.9.0      cachem_1.1.0      parallelly_1.37.1
## [21] tidyselect_1.2.1  digest_0.6.35     stringi_1.8.4     listenv_0.9.1
## [25] rprojroot_2.0.4   fastmap_1.2.0     grid_4.4.0        colorspace_2.1-0
## [29] cli_3.6.2         magrittr_2.0.3    utf8_1.2.4        withr_3.0.0
## [33] scales_1.3.0      backports_1.5.0   bit64_4.0.5       timechange_0.3.0
## [37] rmarkdown_2.27    globals_0.16.3    bit_4.0.5         hms_1.1.3
## [41] memoise_2.0.1     evaluate_0.24.0   viridisLite_0.4.2 rlang_1.1.4
## [45] xml2_1.3.6        svglite_2.1.3     rstudioapi_0.16.0 R6_2.5.1
## [49] systemfonts_1.1.0
```

Analyze attack trends

```
data_dir <- here(glue("{params$data}/{params$simulation}/results"))

success_fnames <-
  dir_ls(data_dir, glob = glue("*norm_{params$norm}*.csv"))

stopifnot(length(success_fnames) == 960)

# every fname is a simulation
success_raw_data <- get_data(success_fnames, read_csv) |>
  glimpse()
```

```
## Rows: 960
## Columns: 18
## $ fname                 <chr> "/Users/zbli/Documents/Documents - ZhaoBin's M~
## $ num_iteration         <dbl> 200, 200, 200, 200, 200, 200, 200, 200, 200, 2~
## $ max_norm              <dbl> 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05~
## $ model_name            <ord> Cascade R-CNN, Faster R-CNN, RetinaNet, SSD, Y~
## $ loss_target           <ord> Mislabeling, Mislabeling, Mislabeling, Mislabe~
## $ attack_bbox           <chr> "predictions", "predictions", "predictions", "~
## $ perturb_fun           <chr> "perturb_inside", "perturb_inside", "perturb_i~
## $ sample_count          <dbl> 52, 52, 52, 52, 53, 52, 52, 52, 52, 53, 52, 52~
## $ attack_count          <dbl> 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50~
## $ success_count         <dbl> 10, 10, 5, 2, 21, 13, 13, 12, 7, 13, 15, 16, 1~
## $ vanish_count          <dbl> 3, 2, 1, 1, 10, 10, 13, 8, 5, 11, 14, 14, 18, ~
## $ mislabel_count        <dbl> 7, 8, 4, 1, 11, 3, 0, 4, 2, 2, 1, 2, 1, 1, 0, ~
```

```
## $ mislabel_intended_count <dbl> 7, 8, 4, 1, 11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ target_max_conf         <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ perturb_min_size        <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ bbox_max_dist           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ bbox_length             <dbl> 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0~
## $ boundary_distance       <dbl> 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01~
```

```
# expand success per simulation into 1 and 0s per row
success_expanded_data <- success_raw_data |>
  rename(
    bbox_dist = boundary_distance,
    bbox_len = bbox_length
  ) |>
  rowwise() |>
  mutate(success = list(rep(0:1, times = c(attack_count - success_count, success_count)))) |>
  unnest_longer(success) |>
  glimpse()
```

```
## Rows: 48,000
## Columns: 19
## $ fname                   <chr> "/Users/zbli/Documents/Documents - ZhaoBin's M~
## $ num_iteration           <dbl> 200, 200, 200, 200, 200, 200, 200, 200, 200, 2~
## $ max_norm                <dbl> 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05, 0.05~
## $ model_name              <ord> Cascade R-CNN, Cascade R-CNN, Cascade R-CNN, C~
## $ loss_target             <ord> Mislabeling, Mislabeling, Mislabeling, Mislabe~
## $ attack_bbox             <chr> "predictions", "predictions", "predictions", "~
## $ perturb_fun             <chr> "perturb_inside", "perturb_inside", "perturb_i~
## $ sample_count            <dbl> 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52, 52~
## $ attack_count            <dbl> 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50~
## $ success_count           <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10~
## $ vanish_count            <dbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3~
## $ mislabel_count          <dbl> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7~
## $ mislabel_intended_count <dbl> 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7~
## $ target_max_conf         <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ perturb_min_size        <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ bbox_max_dist           <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ bbox_len                <dbl> 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0~
## $ bbox_dist               <dbl> 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01~
## $ success                 <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```
# control both
model <- partial(glm_model, predictor = "bbox_dist * bbox_len")
data <- success_expanded_data

reg_res <- get_tidied_reg(model, data, return_mod = TRUE)
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'model_name', 'loss_target'. You can
## override using the `.groups` argument.
```

```r
reg_est <- reg_res$tidied

ext_sig(reg_est, "neg", "bbox_dist")
```

```
## ----------bbox_dist----------
## Total 15 predictors:
## 15 (100%) significant;
## 15 (100%) neg
```

```
## # A tibble: 15 x 9
## # Groups:   model_name, loss_target [15]
##    model_name    loss_target term  estimate std.error statistic p.value conf.low
##    <ord>         <ord>       <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 YOLOv3        Vanishing   bbox~    -6.05      1.23     -4.93       0    -8.47
##  2 YOLOv3        Mislabeling bbox~    -7.15      1.23     -5.81       0    -9.59
##  3 YOLOv3        Untargeted  bbox~    -9.32      1.52     -6.15       0   -12.3
##  4 SSD           Vanishing   bbox~   -10.4       1.55     -6.71       0   -13.5
##  5 SSD           Mislabeling bbox~    -8.00      1.70     -4.71       0   -11.4
##  6 SSD           Untargeted  bbox~    -9.78      1.87     -5.23       0   -13.5
##  7 RetinaNet     Vanishing   bbox~   -23.0       3.08     -7.48       0   -29.3
##  8 RetinaNet     Mislabeling bbox~   -22.5       4.24     -5.32       0   -31.3
##  9 RetinaNet     Untargeted  bbox~   -23.5       2.44     -9.64       0   -28.4
## 10 Faster R-CNN  Vanishing   bbox~   -31.8       4.00     -7.95       0   -39.9
## 11 Faster R-CNN  Mislabeling bbox~   -28.0       4.33     -6.47       0   -36.9
## 12 Faster R-CNN  Untargeted  bbox~   -29.7       2.76    -10.8        0   -35.3
## 13 Cascade R-CNN Vanishing   bbox~   -33.2       4.28     -7.76       0   -41.9
## 14 Cascade R-CNN Mislabeling bbox~   -34.8       5.23     -6.65       0   -45.6
## 15 Cascade R-CNN Untargeted  bbox~   -45.7       4.12    -11.1        0   -54.0
## # i 1 more variable: conf.high <dbl>
```

```r
ext_sig(reg_est, "pos", "bbox_len")
```

```
## ----------bbox_len----------
## Total 15 predictors:
## 15 (100%) significant;
## 15 (100%) pos
```

```
## # A tibble: 15 x 9
## # Groups:   model_name, loss_target [15]
##    model_name    loss_target term  estimate std.error statistic p.value conf.low
##    <ord>         <ord>       <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 YOLOv3        Vanishing   bbox~    8.24      0.53      15.6    0        7.23
##  2 YOLOv3        Mislabeling bbox~    5.89      0.395     14.9    0        5.13
##  3 YOLOv3        Untargeted  bbox~    2.06      0.285      7.24   0        1.51
##  4 SSD           Vanishing   bbox~    4.74      0.318     14.9    0        4.12
##  5 SSD           Mislabeling bbox~    6.06      0.345     17.6    0        5.40
##  6 SSD           Untargeted  bbox~    3.80      0.314     12.1    0        3.19
##  7 RetinaNet     Vanishing   bbox~    2.58      0.345      7.49   0        1.91
##  8 RetinaNet     Mislabeling bbox~    1.26      0.419      3.01   0.003    0.443
##  9 RetinaNet     Untargeted  bbox~    2.53      0.334      7.57   0        1.88
## 10 Faster R-CNN  Vanishing   bbox~    2.08      0.36       5.77   0        1.38
## 11 Faster R-CNN  Mislabeling bbox~    0.955     0.395      2.42   0.016    0.185
## 12 Faster R-CNN  Untargeted  bbox~    1.49      0.312      4.78   0        0.886
## 13 Cascade R-CNN Vanishing   bbox~    3.93      0.405      9.71   0        3.14
## 14 Cascade R-CNN Mislabeling bbox~    1.85      0.395      4.70   0        1.08
```

```
## 15 Cascade R-CNN Untargeted  bbox~    0.675     0.327      2.06   0.039    0.036
## # i 1 more variable: conf.high <dbl>
```

```
ext_sig(reg_est, "both", "bbox_dist:bbox_len")
```

```
## ----------bbox_dist:bbox_len----------
## Total 15 predictors:
## 7 (47%) significant;
## 7 (47%) both
```

```
## # A tibble: 7 x 9
## # Groups:   model_name, loss_target [7]
##   model_name    loss_target term   estimate std.error statistic p.value conf.low
##   <ord>         <ord>       <chr>     <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
## 1 YOLOv3        Vanishing   bbox_~   -18.2       3.54     -5.14   0        -25.2
## 2 SSD           Mislabeling bbox_~   -12.7       3.35     -3.77   0        -19.2
## 3 SSD           Untargeted  bbox_~    -7.44      3.64     -2.05   0.041    -14.5
## 4 RetinaNet     Untargeted  bbox_~    37.7       4.19      8.99   0         29.6
## 5 Faster R-CNN  Untargeted  bbox_~    36.7       4.55      8.07   0         27.9
## 6 Cascade R-CNN Vanishing   bbox_~   -22.5       8.93     -2.52   0.012    -40.0
## 7 Cascade R-CNN Untargeted  bbox_~    47.7       6.64      7.19   0         35.0
## # i 1 more variable: conf.high <dbl>
```

```
dist_lab <- "Perturb-Target Distance"
len_lab <- "Perturb Box Length"

pred_name <- glue("{dist_lab} and {len_lab}, both relative to image width or height,")
main_pt <- glue("longer {len_lab} or shorter {dist_lab} cause success rates to significantly increase f

print_statistics(reg_est, table_caption(pred_name, main_pt, "deliberate"))
```

Table 1: We run a logistic model regressing success against perturb-target distance and perturb box length, both relative to image width or height, in the deliberate attack experiment. Longer perturb box length or shorter perturb-target distance cause success rates to significantly increase for all model and attack combinations, except for perturb box length in untargeted attack on Cascade R-CNN. The interaction terms, even when significant, are negligibly close to 0. Table headers are explained in Appendix **??**.

Group	Regression							
Attack	term	sig	estimate	std.error	statistic	p.value	conf.low	conf.high
YOLOv3								
Vanishing	distance	*	-6.047	1.227	-4.928	0.000	-8.472	-3.660
	length	*	8.243	0.530	15.558	0.000	7.227	9.305
	distance * length	*	-18.211	3.543	-5.140	0.000	-25.189	-11.292
Mislabeling	distance	*	-7.151	1.231	-5.810	0.000	-9.588	-4.761
	length	*	5.888	0.395	14.922	0.000	5.126	6.674
	distance * length		-3.239	3.100	-1.045	0.296	-9.296	2.862
Untargeted	distance	*	-9.320	1.515	-6.153	0.000	-12.343	-6.401
	length	*	2.063	0.285	7.245	0.000	1.508	2.624
	distance * length		4.340	2.943	1.475	0.140	-1.392	10.150
SSD								

	Vanishing	distance	*	-10.417	1.552	-6.711	0.000	-13.513	-7.424
		length	*	4.737	0.318	14.882	0.000	4.120	5.368
		distance * length		-3.353	3.072	-1.091	0.275	-9.345	2.705
	Mislabeling	distance	*	-7.996	1.697	-4.712	0.000	-11.385	-4.729
		length	*	6.065	0.345	17.570	0.000	5.397	6.750
		distance * length	*	-12.651	3.354	-3.772	0.000	-19.201	-6.047
	Untargeted	distance	*	-9.777	1.868	-5.233	0.000	-13.530	-6.201
		length	*	3.798	0.314	12.094	0.000	3.188	4.419
		distance * length	*	-7.443	3.635	-2.048	0.041	-14.527	-0.268
RetinaNet									
	Vanishing	distance	*	-23.008	3.077	-7.477	0.000	-29.253	-17.194
		length	*	2.583	0.345	7.491	0.000	1.912	3.264
		distance * length		-10.769	6.353	-1.695	0.090	-23.153	1.757
	Mislabeling	distance	*	-22.522	4.237	-5.316	0.000	-31.273	-14.667
		length	*	1.261	0.419	3.007	0.003	0.443	2.087
		distance * length		1.459	8.334	0.175	0.861	-14.680	18.011
	Untargeted	distance	*	-23.500	2.437	-9.643	0.000	-28.382	-18.828
		length	*	2.528	0.334	7.571	0.000	1.880	3.189
		distance * length	*	37.697	4.191	8.994	0.000	29.615	46.048
Faster R-CNN									
	Vanishing	distance	*	-31.756	3.996	-7.947	0.000	-39.875	-24.217
		length	*	2.075	0.360	5.770	0.000	1.375	2.785
		distance * length		-0.099	7.820	-0.013	0.990	-15.305	15.352
	Mislabeling	distance	*	-28.038	4.331	-6.474	0.000	-36.927	-19.955
		length	*	0.955	0.395	2.419	0.016	0.185	1.734
		distance * length		10.044	8.211	1.223	0.221	-5.864	26.342
	Untargeted	distance	*	-29.741	2.761	-10.770	0.000	-35.304	-24.477
		length	*	1.494	0.312	4.783	0.000	0.886	2.111
		distance * length	*	36.707	4.548	8.071	0.000	27.946	45.780
Cascade R-CNN									
	Vanishing	distance	*	-33.193	4.280	-7.755	0.000	-41.863	-25.092
		length	*	3.929	0.405	9.706	0.000	3.145	4.732
		distance * length	*	-22.519	8.925	-2.523	0.012	-39.964	-4.967
	Mislabeling	distance	*	-34.815	5.234	-6.652	0.000	-45.560	-25.047
		length	*	1.853	0.395	4.698	0.000	1.085	2.632
		distance * length		-2.173	10.288	-0.211	0.833	-22.101	18.246
	Untargeted	distance	*	-45.652	4.120	-11.080	0.000	-53.998	-37.841
		length	*	0.675	0.327	2.061	0.039	0.036	1.320
		distance * length	*	47.723	6.636	7.191	0.000	34.958	60.993

```r
reg_mod <- reg_res$mod

newdata <- expand_grid(
  bbox_dist = linear_space(data$bbox_dist),
  bbox_len = unique(data$bbox_len)
) |>
  glimpse()
```

```
## Rows: 400
## Columns: 2
## $ bbox_dist <dbl> 0.01000000, 0.01000000, 0.01000000, 0.01000000, 0.01191919, ~
## $ bbox_len  <dbl> 0.1, 0.3, 0.5, 0.7, 0.1, 0.3, 0.5, 0.7, 0.1, 0.3, 0.5, 0.7, ~
```

```r
# type.predict = "link" by default
# https://broom.tidymodels.org/reference/augment.glm.html
# https://stackoverflow.com/questions/14423325/confidence-intervals-for-predictions-from-logistic-regre
reg_pred <- reg_mod |>
  summarize(augment(mod, newdata = newdata, se_fit = TRUE)) |>
  mutate(success = plogis(.fitted), ul = plogis(.fitted + 1.96 * .se.fit), ll = plogis(.fitted - 1.96 *
  glimpse()
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'model_name', 'loss_target'. You can
## override using the `.groups` argument.
```

```
## Rows: 6,000
## Columns: 9
## Groups: model_name, loss_target [15]
## $ model_name  <ord> YOLOv3, YOLOv3, YOLOv3, YOLOv3, YOLOv3, YOLOv3, YOLOv3, YO~
## $ loss_target <ord> Vanishing, Vanishing, Vanishing, Vanishing, Vanishing, Van~
## $ bbox_dist   <dbl> 0.01000000, 0.01000000, 0.01000000, 0.01000000, 0.01191919~
## $ bbox_len    <dbl> 0.1, 0.3, 0.5, 0.7, 0.1, 0.3, 0.5, 0.7, 0.1, 0.3, 0.5, 0.7~
## $ .fitted     <dbl> 0.2227890, 1.8349264, 3.4470637, 5.0592010, 0.2076889, 1.8~
## $ .se.fit     <dbl> 0.10111954, 0.08791323, 0.15870232, 0.25019639, 0.09975179~
## $ success     <dbl> 0.5554680, 0.8623476, 0.9691435, 0.9936894, 0.5517364, 0.8~
## $ ul          <dbl> 0.6037185, 0.8815548, 0.9772042, 0.9961260, 0.5994568, 0.8~
## $ ll          <dbl> 0.5061484, 0.8405889, 0.9583538, 0.9897362, 0.5030438, 0.8~
```

```r
arb_cap <- glue("{emp_tex('Perturbing an arbitrary region obfuscates intent with increased success for a

arb_cap
```

```
## Perturbing an arbitrary region obfuscates intent with increased success for all models and attacks ev
```

```r
g <- success_expanded_data |> ggplot(aes(bbox_dist, success, color = bbox_len, group = bbox_len)) +
  stat_summary(fun.data = "mean_cl_boot") +
  facet_grid(cols = vars(model_name), rows = vars(loss_target))

# https://github.com/tidyverse/ggplot2/blob/ef00be7e2016e1259b4aef7f7c85651df123beff/R/geom-smooth.r#L1
g <- g + geom_ribbon(
```

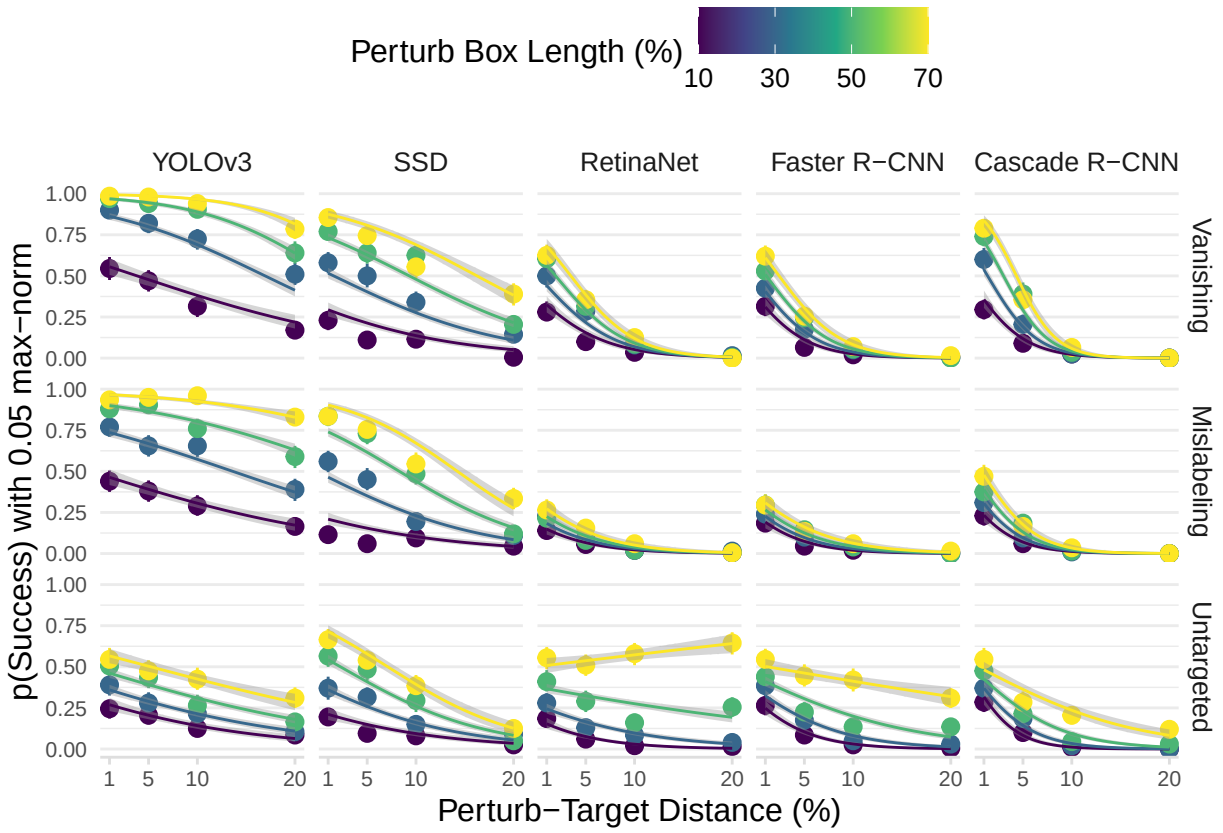

Figure 1: Perturbing an arbitrary region obfuscates intent with increased success for all models and attacks even with 0.05 max-norm: We implement intent obfuscating attack by perturbing an arbitrary non-overlapping square region to disrupt a randomly selected target object at various lengths and distances. The binned summaries and regression trendlines graph success proportion against perturb-target distance and perturb box length, both relative to image width or height, in the deliberate attack experiment. Errors are 95% confidence intervals and every point aggregates success over 200 images. The deliberate attack multiplies success as compared to the randomized attack (Figure **??**), especially at close perturb-target distance and large perturb box length. Full details are given in Section **??**.

```
  data = reg_pred, aes(ymin = ll, ymax = ul),
  fill = "grey60", linetype = 0, alpha = 0.4
) +
  geom_line(data = reg_pred)

g + labs(x = glue("{dist_lab} (%)"), y = glue("p(Success) {norm_axy(params$norm)}")) +
  scale_x_continuous(breaks = unique(success_expanded_data$bbox_dist), labels = scales::label_percent(su
  scale_color_viridis_c(name = glue("{len_lab} (%)"), breaks = unique(success_expanded_data$bbox_len),
```

```
get_reg_vars <- function(data) {
  data |> select(bbox_dist, bbox_size_perturb, model_name, loss_target, success, object)
}

# run random.Rmd 1st
rand_dist_size <- readRDS(here(glue("analysis/rand_dist_size_norm_{params$norm}.RDS"))) |>
  mutate(object = 1) |>
  get_reg_vars() |>
```

```
  glimpse()
```

```
## Rows: 60,000
## Columns: 6
## $ bbox_dist         <dbl> 0.43469769, 0.58799145, 0.16133960, 0.16319737, 0.28~
## $ bbox_size_perturb <dbl> 0.0021062328, 0.0034026994, 0.0392784101, 0.06702487~
## $ model_name        <ord> Cascade R-CNN, Cascade R-CNN, Cascade R-CNN, Cascade~
## $ loss_target       <ord> Mislabeling, Mislabeling, Mislabeling, Mislabeling, ~
## $ success           <dbl> 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ object            <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
```

```
comb_dist_size <- success_expanded_data |>
  mutate(object = 0, bbox_size_perturb = bbox_len^2) |>
  get_reg_vars() |>
  bind_rows(rand_dist_size) |>
  mutate(
    bbox_dist = bbox_dist,
    bbox_size_perturb = bbox_size_perturb
  ) |>
  glimpse()
```

```
## Rows: 108,000
## Columns: 6
## $ bbox_dist         <dbl> 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01~
## $ bbox_size_perturb <dbl> 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 0.01~
## $ model_name        <ord> Cascade R-CNN, Cascade R-CNN, Cascade R-CNN, Cascade~
## $ loss_target       <ord> Mislabeling, Mislabeling, Mislabeling, Mislabeling, ~
## $ success           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ object            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

```
stopifnot(nrow(comb_dist_size) == nrow(success_expanded_data) +
  nrow(rand_dist_size) && sum(is.na(comb_dist_size)) == 0)
```

```
# control both
model <- partial(glm_model, predictor = "object + bbox_dist * bbox_size_perturb")
data <- comb_dist_size

reg_est <- get_tidied_reg(model, data)
```

```
## Warning: There were 5 warnings in `mutate()`.
## The first warning was:
## i In argument: `mod = list(model(data))`.
## i In row 8.
## Caused by warning:
## ! glm.fit: fitted probabilities numerically 0 or 1 occurred
## i Run `dplyr::last_dplyr_warnings()` to see the 4 remaining warnings.
```

```
## Warning: There were 221 warnings in `summarize()`.
## The first warning was:
## i In argument: `tidy_plus_plus(mod, conf.int = TRUE)`.
## i In row 7.
## Caused by warning:
## ! glm.fit: fitted probabilities numerically 0 or 1 occurred
## i Run `dplyr::last_dplyr_warnings()` to see the 220 remaining warnings.
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
```

```
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'model_name', 'loss_target'. You can
## override using the `.groups` argument.
```

```
ext_sig(reg_est, "neg", "object")
```

```
## ----------object----------
## Total 15 predictors:
## 13 (87%) significant;
## 12 (80%) neg
```

```
## # A tibble: 12 x 9
## # Groups:   model_name, loss_target [12]
##    model_name    loss_target term  estimate std.error statistic p.value conf.low
##    <ord>         <ord>       <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 YOLOv3        Vanishing   obje~   -0.652     0.068     -9.55   0       -0.786
##  2 YOLOv3        Mislabeling obje~   -0.609     0.065     -9.41   0       -0.736
##  3 YOLOv3        Untargeted  obje~   -0.754     0.081     -9.28   0       -0.915
##  4 RetinaNet     Vanishing   obje~   -0.5       0.092     -5.45   0       -0.681
##  5 RetinaNet     Mislabeling obje~   -0.444     0.13      -3.41   0.001   -0.703
##  6 RetinaNet     Untargeted  obje~   -0.337     0.085     -3.97   0       -0.504
##  7 Faster R-CNN  Vanishing   obje~   -1.02      0.117     -8.66   0       -1.25
##  8 Faster R-CNN  Mislabeling obje~   -0.999     0.146     -6.87   0       -1.29
##  9 Faster R-CNN  Untargeted  obje~   -0.497     0.093     -5.32   0       -0.682
## 10 Cascade R-CNN Vanishing   obje~   -0.921     0.105     -8.77   0       -1.13
## 11 Cascade R-CNN Mislabeling obje~   -0.851     0.127     -6.71   0       -1.10
## 12 Cascade R-CNN Untargeted  obje~   -0.615     0.101     -6.08   0       -0.815
## # i 1 more variable: conf.high <dbl>
```

```
ext_sig(reg_est, "neg", "bbox_dist")
```

```
## ----------bbox_dist----------
## Total 15 predictors:
## 15 (100%) significant;
## 15 (100%) neg
```

```
## # A tibble: 15 x 9
## # Groups:   model_name, loss_target [15]
##    model_name    loss_target term  estimate std.error statistic p.value conf.low
##    <ord>         <ord>       <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 YOLOv3        Vanishing   bbox~    -8.41     0.505     -16.6       0    -9.42
##  2 YOLOv3        Mislabeling bbox~    -8.34     0.481     -17.3       0    -9.30
##  3 YOLOv3        Untargeted  bbox~   -11.6      0.794     -14.6       0   -13.2
##  4 SSD           Vanishing   bbox~   -14.8      0.721     -20.6       0   -16.3
##  5 SSD           Mislabeling bbox~   -14.8      0.815     -18.1       0   -16.4
##  6 SSD           Untargeted  bbox~   -15.9      0.944     -16.8       0   -17.8
##  7 RetinaNet     Vanishing   bbox~   -28.6      1.82      -15.7       0   -32.2
##  8 RetinaNet     Mislabeling bbox~   -30.9      2.84      -10.9       0   -36.7
##  9 RetinaNet     Untargeted  bbox~   -14.1      1.03      -13.7       0   -16.2
## 10 Faster R-CNN  Vanishing   bbox~   -27.9      2.15      -13.0       0   -32.2
## 11 Faster R-CNN  Mislabeling bbox~   -23.2      2.26      -10.3       0   -27.8
```

```
## 12 Faster R-CNN  Untargeted  bbox~   -18.7      1.24      -15.2       0   -21.2
## 13 Cascade R-CNN Vanishing   bbox~   -35.3      2.41      -14.6       0   -40.2
## 14 Cascade R-CNN Mislabeling bbox~   -32.4      2.95      -11.0       0   -38.5
## 15 Cascade R-CNN Untargeted  bbox~   -29.5      1.92      -15.4       0   -33.4
## # i 1 more variable: conf.high <dbl>
```

```
ext_sig(reg_est, "pos", "bbox_size_perturb")
```

```
## ----------bbox_size_perturb----------
## Total 15 predictors:
## 15 (100%) significant;
## 15 (100%) pos
```

```
## # A tibble: 15 x 9
## # Groups:   model_name, loss_target [15]
##    model_name    loss_target term  estimate std.error statistic p.value conf.low
##    <ord>         <ord>       <chr>    <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
##  1 YOLOv3        Vanishing   bbox~    16.6      0.869     19.1    0       14.9
##  2 YOLOv3        Mislabeling bbox~     8.67     0.501     17.3    0        7.71
##  3 YOLOv3        Untargeted  bbox~     1.81     0.29       6.25   0        1.25
##  4 SSD           Vanishing   bbox~     5.34     0.337     15.9    0        4.68
##  5 SSD           Mislabeling bbox~     5.48     0.333     16.5    0        4.84
##  6 SSD           Untargeted  bbox~     3.26     0.295     11.1    0        2.69
##  7 RetinaNet     Vanishing   bbox~     2.81     0.351      8.01   0        2.13
##  8 RetinaNet     Mislabeling bbox~     0.99     0.434      2.28   0.022    0.141
##  9 RetinaNet     Untargeted  bbox~     3.06     0.3       10.2    0        2.48
## 10 Faster R-CNN  Vanishing   bbox~     2.87     0.392      7.32   0        2.11
## 11 Faster R-CNN  Mislabeling bbox~     1.06     0.426      2.49   0.013    0.225
## 12 Faster R-CNN  Untargeted  bbox~     1.82     0.308      5.92   0        1.22
## 13 Cascade R-CNN Vanishing   bbox~     5.07     0.428     11.8    0        4.24
## 14 Cascade R-CNN Mislabeling bbox~     2.85     0.398      7.15   0        2.07
## 15 Cascade R-CNN Untargeted  bbox~     1.31     0.322      4.08   0        0.682
## # i 1 more variable: conf.high <dbl>
```

```
ext_sig(reg_est, "both", "bbox_dist:bbox_size_perturb")
```

```
## ----------bbox_dist:bbox_size_perturb----------
## Total 15 predictors:
## 8 (53%) significant;
## 8 (53%) both
```

```
## # A tibble: 8 x 9
## # Groups:   model_name, loss_target [8]
##   model_name    loss_target term   estimate std.error statistic p.value conf.low
##   <ord>         <ord>       <chr>     <dbl>     <dbl>     <dbl>   <dbl>    <dbl>
## 1 YOLOv3        Vanishing   bbox_~    -47.8      4.88     -9.80   0       -57.5
## 2 YOLOv3        Mislabeling bbox_~    -10.4      3.25     -3.20   0.001   -16.8
## 3 YOLOv3        Untargeted  bbox_~     14.5      2.74      5.28   0         9.12
## 4 RetinaNet     Mislabeling bbox_~     20.0      8.75      2.28   0.022     2.68
## 5 RetinaNet     Untargeted  bbox_~     32.8      3.06     10.7    0        26.9
## 6 Faster R-CNN  Untargeted  bbox_~     35.1      3.35     10.5    0        28.6
## 7 Cascade R-CNN Vanishing   bbox_~    -37.9      9.34     -4.06   0       -56.5
## 8 Cascade R-CNN Untargeted  bbox_~     33.6      5.11      6.58   0        23.7
## # i 1 more variable: conf.high <dbl>
```

```
dist_lab <- "Perturb-Target Distance"
size_lab <- "Perturb Box Size"
```

```
pred_name <- glue("object (versus non-object), with {dist_lab} and {size_lab} as covariates, both relati
main_pt <- "perturbing an object (in the randomized attack) rather than a non-object (in the deliberate

tab_cap <- glue("We combined the data in the randomized and deliberate attack experiments to run a logis

print_statistics(reg_est, tab_cap)
```

Table 2: We combined the data in the randomized and deliberate attack experiments to run a logistic model regressing success against object (versus non-object), with perturb-target distance and perturb box size as covariates, both relative to image width or height. The "object" term codes object as 1 and non-object as 0. Perturbing an object (in the randomized attack) rather than a non-object (in the deliberate attack) significantly decreases success rates for all model and attack combinations, after controlling for perturb sizes and perturb-target distances. Table headers are explained in Appendix **??**.

Group	Regression							
Attack	term	sig	estimate	std.error	statistic	p.value	conf.low	conf.high
YOLOv3								
Vanishing	object	*	-0.652	0.068	-9.554	0.000	-0.786	-0.518
	distance	*	-8.411	0.505	-16.639	0.000	-9.417	-7.435
	size	*	16.598	0.869	19.097	0.000	14.934	18.341
	distance * size	*	-47.808	4.879	-9.798	0.000	-57.511	-38.378
Mislabeling	object	*	-0.609	0.065	-9.412	0.000	-0.736	-0.482
	distance	*	-8.339	0.481	-17.321	0.000	-9.298	-7.411
	size	*	8.671	0.501	17.299	0.000	7.708	9.674
	distance * size	*	-10.405	3.246	-3.205	0.001	-16.804	-4.076
Untargeted	object	*	-0.754	0.081	-9.276	0.000	-0.915	-0.596
	distance	*	-11.625	0.794	-14.649	0.000	-13.213	-10.102
	size	*	1.812	0.290	6.251	0.000	1.245	2.381
	distance * size	*	14.484	2.744	5.278	0.000	9.123	19.884
SSD								
Vanishing	object	*	0.246	0.068	3.621	0.000	0.113	0.379
	distance	*	-14.818	0.721	-20.550	0.000	-16.258	-13.431
	size	*	5.337	0.337	15.851	0.000	4.685	6.005
	distance * size		3.676	2.807	1.309	0.190	-1.832	9.177
Mislabeling	object		-0.066	0.071	-0.939	0.348	-0.205	0.072
	distance	*	-14.788	0.815	-18.149	0.000	-16.417	-13.223
	size	*	5.484	0.333	16.489	0.000	4.839	6.143
	distance * size		0.065	2.988	0.022	0.983	-5.800	5.917
Untargeted	object		-0.004	0.075	-0.060	0.952	-0.151	0.142
	distance	*	-15.892	0.944	-16.828	0.000	-17.786	-14.084
	size	*	3.265	0.295	11.078	0.000	2.691	3.847
	distance * size		5.899	3.236	1.823	0.068	-0.458	12.233

RetinaNet								
Vanishing	object	*	-0.500	0.092	-5.453	0.000	-0.681	-0.321
	distance	*	-28.569	1.820	-15.698	0.000	-32.231	-25.097
	size	*	2.814	0.351	8.006	0.000	2.130	3.509
	distance * size		4.598	6.024	0.763	0.445	-7.315	16.315
Mislabeling	object	*	-0.444	0.130	-3.413	0.001	-0.703	-0.192
	distance	*	-30.944	2.839	-10.901	0.000	-36.733	-25.604
	size	*	0.990	0.434	2.282	0.022	0.141	1.842
	distance * size	*	19.990	8.749	2.285	0.022	2.681	37.028
Untargeted	object	*	-0.337	0.085	-3.972	0.000	-0.504	-0.171
	distance	*	-14.091	1.029	-13.692	0.000	-16.158	-12.124
	size	*	3.062	0.300	10.202	0.000	2.475	3.652
	distance * size	*	32.836	3.065	10.714	0.000	26.912	38.930
Faster R-CNN								
Vanishing	object	*	-1.017	0.117	-8.659	0.000	-1.250	-0.790
	distance	*	-27.903	2.147	-12.997	0.000	-32.247	-23.831
	size	*	2.870	0.392	7.320	0.000	2.107	3.644
	distance * size		-13.406	7.814	-1.716	0.086	-28.896	1.751
Mislabeling	object	*	-0.999	0.146	-6.866	0.000	-1.291	-0.720
	distance	*	-23.220	2.264	-10.254	0.000	-27.839	-18.962
	size	*	1.062	0.426	2.490	0.013	0.225	1.897
	distance * size		2.781	7.934	0.350	0.726	-13.013	18.122
Untargeted	object	*	-0.497	0.093	-5.319	0.000	-0.682	-0.315
	distance	*	-18.743	1.235	-15.173	0.000	-21.226	-16.383
	size	*	1.824	0.308	5.925	0.000	1.222	2.429
	distance * size	*	35.146	3.351	10.487	0.000	28.643	41.785
Cascade R-CNN								
Vanishing	object	*	-0.921	0.105	-8.773	0.000	-1.128	-0.717
	distance	*	-35.343	2.414	-14.641	0.000	-40.213	-30.751
	size	*	5.069	0.428	11.831	0.000	4.242	5.923
	distance * size	*	-37.908	9.338	-4.060	0.000	-56.487	-19.857
Mislabeling	object	*	-0.851	0.127	-6.708	0.000	-1.103	-0.605
	distance	*	-32.428	2.950	-10.994	0.000	-38.452	-26.886
	size	*	2.849	0.398	7.151	0.000	2.071	3.633
	distance * size		-14.661	10.223	-1.434	0.152	-34.890	5.215
Untargeted	object	*	-0.615	0.101	-6.082	0.000	-0.815	-0.419
	distance	*	-29.517	1.920	-15.376	0.000	-33.393	-25.864
	size	*	1.311	0.322	4.077	0.000	0.682	1.943
	distance * size	*	33.612	5.105	6.584	0.000	23.655	43.689