AN MODEL GENERALIZATION STUDY IN LOCALIZING INDOOR COWS WITH COW LOCALIZATION (COLO) DATASET

A PREPRINT

Mautushi Das
School of Animal Sciences
Virginia Tech
Blacksburg, VA 24061
mautushid@vt.edu

Gonzalo Ferreira
School of Animal Sciences
Virginia Tech
Blacksburg, VA 24061
gonf@vt.edu

C. P. James Chen *
School of Animal Sciences
Virginia Tech
Blacksburg, VA 24061
niche@vt.edu

May 13, 2024

ABSTRACT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Keywords Object detection · Model selection · Model generalization

*Corresponding author: James Chen <niche@vt.edu>

1 Introduction

Define object localization and applications

Localizing livestock individuals from images or videos has became an essential task in precision livestock farming (PLF) []. Such techniques allows researchers and farm managers to monitor the health and well-being of animals in real-time, optimizing their resource management and improving sustainability []. Technically speaking, in the field of computer vision (CV), which is a subfield of artificial intelligence (AI) that focuses on translating visual information into actionable insights, the localization tasks can be further categorized into object detection, object segmentation, and pose estimation. Object detection is the simplest form of localization, which localizes objects of interest by enclosing them within a rectangular bounding box defined by x and y coordinates, pixel width, and pixel height. <examples 1, 2>. To have a finer localization, object segmentation is deployed to outline the object contours pixel-wise, while pose estimation is achieved by orienting and marking the keypoints of the object. <another examples 3, 4>.

Model generalization, pre-training, and fine-tuning

Although implementing image-based systems in the livestock production has shown promising results, current studies merely focus on the accuracy on homogenous environments and rarely address the challenges of model generalization. Model generalization refers to how well a model can perform on unseen data, which is crucial when ones want to reproduce existing studies or models in their own environments. The generalization of a CV model can be affected by a variety of factors in the deployment environment, such as camera angles and the presence of occlusions. Deploying the same model in a new environment with different conditions cannot necessarily guarantee the same performance as reported in the original study. [] Li2021 also pointed out that lightning condition of farms in real applications can be highly variable, leading a poor generalization th new environments.

One explaination for the poor generalization is the discrepancy between the pre-training process and the specific use case. Most CV models are released with pre-trained weights, which were obtained from the results of training on a large-scale dataset. For example, the COCO dataset [] is a general-purpose dataset that contains more than 200 thousands images and a wide range of object categories, such as vehicles and household items. Directly deploying a model pre-trained on the COCO dataset to specifically detect cows in a farm setting may not ensure satisfactory performance, as the dataset does not contain enough cow instances in different view angles or occlusions. To alleviate the discrepancy, fine-tuning is a common practice that modifies the prediction head of the pre-trained model and updates the weights on a new dataset that is more relevant to the specific use case. Most application studies have adopted this approach to improve the model generalization on their specific tasks (examples 1-5).

Nevertheless, the fine-tuning is not guaranteed to be successful, as the outcome depends on both the quantity and quality of the annotated dataset. For example, zin et. al.(2020) deployed an object detection model to recognize cow ear tags in a dairy farm. Although the model achieved a high accuracy of 92.5% on recognizing the digits on the ear tags, more than 10 thousand images were required for fine-tuning the model. Assemblying such a large dataset is labor-intensive

and requires specific training in annotating the images. Because the annotated dataset is rigorously organized in specific format. For example, the COCO annotation format [] store the image information, object class, and annotations of the entire dataset in one nested JSON format []. Whereas the YOLO format [], another common format for object localization, stores information of one image in one text file, with each line representing one object instance in the image. Additionally, unlike the COCO format that stores bounding box coordinates in absolute pixel values, the YOLO format stores the coordinates in relative values to the image size. These technical details are keys to valid annotations, which are usually helped by the professional annotation tools such as labelme [], CVAT [], or Roboflow [].

Model Complexity and Performance

Another perspective that affects the model generalization is model complexity. In general, model complexity is quantified by the number of learnable parameters in a model []. A more complex model often can better generalize to unseen data with higy accuracy. However, such high complexity also comes with a cost of computational resources in a form of either memory or time []. The computational cost may further limit how the models can be deployed in real-world applications, where real-time processing or edge computing is desired for fast or compact systems. For instance, the VGG-16 model simonyan2014very has 138 million parameters and recommends a video memory of at least 8GB, while the ResNet-152 he2016deep has around 60 million parameters with a recommended video memory of 11GB. Additionally, recent models for object detection such as YOLOv8 [] and YOLOv9 [] have been developed in different sizes and therefore provide a flexible choice for researchers to balance between the generalization performance and the computational cost. In YOLOv8, the spectrum of model complexity ranges from the highly intricate, such as YOLOv8x containing 68.2 million parameters, to more streamlined variants YOLOv8n with only 3.2 million parameters. And the demand for the memory, solely from the model architecture without considering the intermediate results during the training or inferenecne process, is larger in a factor of 21 for YOLOv8x (136.9 megabytes) compared to YOLOv8n (6.5 megabytes). Therefore, the trade-off between the model complexity and the computational cost is a critical factor to consider in deploying CV models in real-world scenarios.

YOLO Models

Public Datasets

A public dataset helps the community to develop methodology based on the same baseline. One famous example in computer vision is the ImageNet dataset [], which serves as a benchmark for image classification. AlexNet [], the winner of the ImageNet Large Scale Visual Recognition Challenge in 2012, show its outstanding capability to classify images in ImageNet dataset using Rectified Linear Units (ReLU) as the activation function than the traditional sigmoid function. The success of AlexNet accelerate the developement of CV models in the following years, such as VGG [], GoogLeNet [], ResNet [], and DenseNet []. However, similar to the challenges that pre-trained models face in the specific use case, a generic public dataset, such as ImageNet and COCO, may not be sufficient to PLF applications.

There were efforts to create public datasets for livestock scenario. For example, XXX [] was collected for xxx. Another example in pigs xxx.

Study Objectives

This study aims to explore model generalization across varying environmental settings and model complexities within the context of indoor cow localization. It seeks to exmaine three practical hypotheses:

- **Model generalization is equally influenced by changes in lighting conditions and camera angles.** Should camera angles prove more impactful than lighting conditions, it would be advisable to prioritize camera placement when deploying computer vision (CV) models in new environments.
- **There is a positive correlation between model complexity and generalization performance.** If a highly complex model does not ensure superior performance, future studies might consider adopting less computationally demanding models that still enhance performance.
- **The choice of the initial weights has no significant impact on the model's generalization performance.** If the weights obtained from the fine-tuning process throught a similar context do not improve the transfer learning to the new environment in comparison to the pre-trained weights, the fine-tuning effort may be deemed unnecessary for certain tasks.

To facilitate these investigations, a public dataset named COws LOcalization (COLO) will be developed and made available to the community. The findings of this study are expected to provide practical guidelines for Precision Livestock Farming (PLF) researchers on deploying CV models, considering available resources and anticipated performance.

2 Materials and Methods

Cow Husbandry

The studied cows were housed in a free-stall barn at Virginia Tech Dairy Comlex at Kentland Farm in Virginia, USA. The cow handling and image capturing were conducted following the guidelines and approval of the Virginia Tech Institutional Animal Care and Use Committee (#IACUC xxxxx).

Image Dataset

The images in this study were collected using the Amazon Ring camera model Spotlight Cam Battery Pro (Ring Inc.), which offers a real-time video feed of dairy cows. Three cameras were installed in the barn: two at a height of 3.25 meters (10.66 feet) above ground covering an area of 33.04 square meters (355.67 square feet). One camera provided a top view while the other was angled approximately 40 degrees from the horizontal to offer a side view of the cows. These are hereafter referred to as *the top-view camera* and *the side-view camera*, respectively. A third camera, termed *the external camera*, was set at a lower height of 2.74 meters (9.00 feet) and covered a larger area of 77.63 square

meters (835.56 square feet). Positioned 10 degrees downward from the horizontal, it captured a challenging perspective prone to occlusions among cows.

Images were captured using an unofficial Ring Application Programming Interface (API) [1], configured to record a ten-second video clip every 30 minutes continuously for 14 days. Since the image quality relies on the camera's internet connection, which was occasionally unstable, some images were found to be tearing or unrecognizable. Hence, the resulting dataset was manually curated for consistent quality, comprising 504 images from *the top-view camera*, 500 from *the side-view camera*, and 250 from *the external camera*. These images were futher categorized based on the lighting conditions: for *the top-view camera*, 296 images were captured during daylight, 118 in the evening under artificial lighting, and 90 as near-infrared images without artificial light. From *the side-view camera*, 113 images were taken in the evening, and 97 as near-infrared images. All images from *the external camera* were captured during the day. The image examples were shown xxx.

The image annotations were conducted using an online platform, Roboflow [], to define cow positions in the images. The bounding boxes were manually drawn to enclose the cow contours, providing the coordinates of the top-left corners and the width and height of the boxes. If cows were partially occluded, the invisible parts were infered based on the adjacent visible parts. If the cow position was too far from the camera and make the important body features, such as head, tail, and legs, unrecognizable, the cow was excluded from the dataset. The final annotations were saved in the YOLO format [], where annotations were stored in a text file with one row per cow in the image, each row containing the cow's class, center coordinates, width, and height of the bounding box. The graphical representation of the annotated images was shown in Figure XXX.

Data Split Design

Study 1: Benchmarking Model Generalization Across Different Conditions

To compare the performance drop between different view angles and lighting conditions, we designed a cross-testing strategy where models were trained on one dataset configuration and tested on another. There are four dataset configurations in this study:

Our study starts with the systematic acquisition of image data, focusing on targeted cattle populations within Kentland Farm at Virginia Tech. All animal handling and media recordings were conducted following the guidelines and approval of the Virginia Tech Institutional Animal Care and Use Committee. This initial phase is succeeded by meticulous data processing steps which include the annotation and formatting of the dataset for machine learning applications. Subsequently, we proceed to fine-tune our dataset utilizing a variety of deep learning architectures. For each model, we meticulously calculate a suite of performance metrics.

Building upon the results obtained from these diverse models, we construct a "summary plot." This plot is designed to elucidate the findings related to the second and third questions delineated in Section contribution of our paper. It will visually guide the selection of an optimal model by delineating the relationship between dataset size and achieved

accuracy, as well as the computational cost versus the precision of the models. Through this analytical representation, we aim to furnish a comprehensive tool that aids researchers in making informed decisions when it comes to choosing the most suitable object detection model for their specific requirements in livestock production studies.

Data Preparation

Recognizing the crucial role of lighting conditions on data integrity, we meticulously orchestrated our data gathering operations at assorted intervals throughout the day, specifically: dawn, midday, dusk, and late evening. This methodical approach was paramount in guaranteeing the inclusion of an extensive spectrum of lighting conditions within our dataset, thereby augmenting its diversity and resilience to various environmental challenges.

Moreover, cognizant of the effect camera angles and perspectives have on capturing the full gamut of cattle postures, we varied our image capture process accordingly. This variation not only accounted for the different positions and movements of the cattle but also for the heterogeneous nature of the environment in which they were situated. In addition, we aimed to ensure a broad representation of breeds by including both Jersey and Holstein cows in our dataset, recognizing that breed-specific characteristics could significantly influence the model's performance.

Animal Husbandry

The farm, hosting a diverse bovine community of over 200 individuals from the Jersey and Holstein breeds, served as an exemplary setting for our endeavor. It offered a plethora of varied scenarios and animal interactions, encapsulating the essence of a vibrant and dynamic agricultural environment. This multifaceted setting was critical in establishing a robust and comprehensive dataset reflective of the real-world complexity and variability one would expect in a livestock farming operation.

Data Annotation and Formation

- Talk about how the data was collected using pipes and Amazon Ring Cameras

- How the data is annotated on Roboflow The methodological rigor involved in data preparation is vital for the integrity of our machine learning model's training process. Here is a clear outline of the steps executed in preparing the dataset for cow detection in our investigation:

1. **Frame Extraction:** We utilized a customized Python script for the extraction of frames from the video streams, converting them into a series of static images. To guarantee uniformity across the dataset, we extracted frames at regular one-second intervals. This process yielded 'n' distinct images, which were then allocated to training, validation, and testing datasets for the subsequent stages of our machine learning endeavor.

2. **Annotation with Roboflow:** We uploaded the frames onto Roboflow, a versatile annotation platform. This tool enabled our team to annotate images by meticulously outlining cows with bounding boxes, ensuring that

the AI model can learn to identify the target objects effectively. The annotated frames are exemplified in Figure **??**.

3. **Annotation Format Selection:** Roboflow's robust export options allowed us to obtain annotations in various formats suitable for different model architectures. We primarily opted for COCO and YOLOv5 formats, both widely recognized for their compatibility with state-of-the-art object detection algorithms.

4. **Data Storage and Maintenance:** Post-annotation, we stored the images in the universally accepted JPG format. Accompanying these images, the corresponding annotation files were meticulously cataloged, readying the dataset for the intricate process of model training and subsequent evaluation.

By adhering to these steps, we ensured the creation of a high-quality, standardized dataset poised for deployment in the development of an AI-powered cow detection system, geared towards enhancing the precision and efficiency of livestock management.

Simulation Design

Data Splits

We aim to thoroughly investigate model generalization across diverse conditions within livestock environments, specifically focusing on cattle localization. To achieve this, we meticulously designed and organized our dataset into five distinct configurations, each representing unique conditions under which the cattle were captured. These configurations are critical for evaluating the robustness and adaptability of object detection models, particularly in terms of their ability to generalize from one set of conditions to another. Below, we detail the dataset configurations and the rationale behind our data split strategy.

Dataset Configurations:

1. **Top View:** Images captured from an overhead perspective, providing a comprehensive view of the livestock area.

2. **Side View:** Images taken at a 60-degree (approx.) angle to the ground, offering a profile perspective of the cattle.

3. **Daylight:** Images captured during daylight conditions from both the top and side views, ensuring natural lighting.

4. **Nighttime:** Images obtained during nighttime from both the top and side views, with lighting conditions significantly reduced.

5. **Breed Specific:** A subset of images exclusively featuring the Holstein breed, allowing for breed-specific model training.

Training and Testing Strategy To rigorously assess model generalization, we employed a cross-testing methodology where models were trained on one dataset configuration and tested on another. This approach enabled us to isolate and understand the impact of various factors—such as viewing angle, lighting conditions, and breed variation—on model performance. The specific training and testing scenarios were as follows:

1. **Viewing Angle Generalization:** Models were trained on the Top View dataset and tested on the Side View dataset, and vice versa. This setup assesses the model's ability to adapt to changes in perspective.
2. **Lighting Condition Generalization:** Models trained on Daylight data were tested on Nighttime data to evaluate performance under varying lighting conditions, and vice versa.
3. **Breed Variation Generalization:** Models trained on the Breed Specific (Holstein) dataset were tested on a mixed-breed dataset (Holstein and Jersey), assessing the impact of breed diversity on detection accuracy.
4. **Comprehensive Generalization:** Finally, models were trained on a combination of all dataset configurations to examine overall generalization capabilities across viewing angles, lighting conditions, and breed variations.

This structured approach to data split and testing is designed to provide insights into the extent to which object detection models, trained under specific conditions, can accurately generalize to different, untrained conditions. By systematically varying training and testing datasets, we aim to uncover potential limitations and strengths of current object detection technologies in the context of livestock monitoring, contributing valuable knowledge towards the development of more robust and adaptable solutions in precision agriculture.

Objective 1: How model performance is decomposed by different factors

To investigate the model generalization. Factors such as lightning..

Each data configuration

Objective 2: How a fine-tuned model performed on a new dataset

Model Training and Evaluation

Trianing hyperparameters

- how to cross validation is Design with different sample Size - Iteration - Evaluation metrics

Data Augmentation

Model evaluation and cross validation

3 Results

The organize dataset is published on the huggingface dataset repository. It's organized in two formats: YOLO and COCO

Model performance decomposition

Model size and architecture performance

Fine-tuned versus. Pre-trained model performance

- by data configuration - by sample Size - by model architecture

External evaluation on the new dataset

3.1 Dataset release

4 Discussion

4.1 Why the lighting condition has less impact on the performance drop?

4.2 Is complex model architecture always better?

4.3 How does the model generalization study help in real-world applications?

5 Conclusion

Your conclusion here

Acknowledgments

This was was supported in part by.....

References

[1] Dusty Greif. dgreif/ring, April 2024. original-date: 2018-10-12T22:53:01Z.

6 Headings: first level

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula. See Section 6.

6.1 Headings: second level

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

$$\xi_{ij}(t) = P(x_t = i, x_{t+1} = j|y, v, w; \theta) = \frac{\alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i(t) a_{ij}^{w_t} \beta_j(t+1) b_j^{v_{t+1}}(y_{t+1})} \tag{1}$$

6.1.1 Headings: third level

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Paragraph Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

7 Examples of citations, figures, tables, references

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum

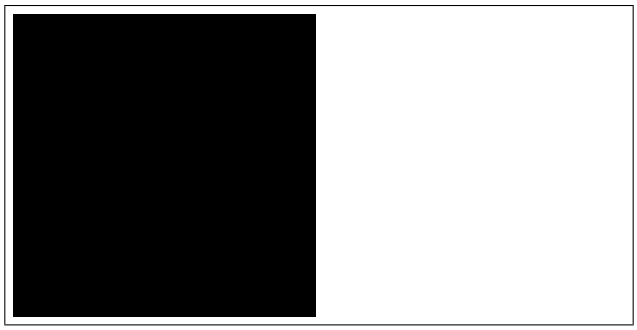

Figure 1: Sample figure caption.

fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui. [**?** **?**] and see [**?**].

The documentation for `natbib` may be found at

`http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf`

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

Hasselmo, et al. (1995) investigated...

`https://www.ctan.org/pkg/booktabs`

7.1 Figures

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi. See Figure 1. Here is how you add footnotes. [2] Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

[2] Sample of the first footnote.

Table 1: Sample table title

	Part	
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

7.2 Tables

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo. See awesome Table 1.

7.3 Lists

- Lorem ipsum dolor sit amet
- consectetur adipiscing elit.
- Aliquam dignissim blandit est, in dictum tortor gravida eget. In ac rutrum magna.