Activation Control of Vision Models for Sustainable AI Systems

Jonathan Burton-Barr, Basura Fernando, and Deepu Rajan

***Abstract*—As AI systems become more complex and widespread, they require significant computational power, increasing energy consumption. Addressing this challenge is essential for ensuring the long-term sustainability of AI technology. AI-on-AI control refers to a system with a set of AI functions controlled by an upper-level AI model. Previous work in AI-on-AI control focuses on boosting accuracy or expanding system capability by increasing overall system cost. Alternatively, we focus on applying AI-on-AI control to decrease system cost and increase the sustainability and viability of a system with multiple AI functions. Our Supervised Image Classification Evaluative Controller (SICEC) is a cost-reduction oriented AI-on-AI controller that learns when vision models within an AI system should be activated based on input features. The function controller preprocesses an input and activates relevant functions, functions being distinct units of AI functionality within the system. Some functions have a set of same functional models. These models take the same input and produce the same output but have architectural differences. We introduce a same functional controller to select a same functional model using the function controller's decision confidence. Results are promising, with a decrease of up to 48.9% in inference time, 67.8% in FLOPs, and 66.4% in energy usage. With SICEC showing significant reductions in inference time and energy cost, our work contributes to limited resource computing and sustainable AI technology.**

***Impact Statement*—Research surrounding AI control mainly focuses on non-AI functions with control over AI functions being rare. Available works use AI control to select AI models for a user-specified AI system or to increase the accuracy of an AI system. When AI control is applied in such a fashion, it can increase energy or inference cost [1][2][3]. Our paper lays foundation for AI-on-AI control that reduces these costs for systems with multiple computer vision functions. SICEC evaluates an input and only activates the relevant system functions for that input. SICEC also attempts to gauge the complexity of an image and assign lower-cost function-related models when possible. Results promote the viability of cost-reductive AI-on-AI control research showing significant energy and inference time reductions. SICEC like methodology could be applied to increase the long-term sustainability of various AI systems, examples being computer vision cloud, GPT constructed, and multi-model robotic systems.**

***Index Terms*—Artificial intelligence in computational sustainability, classification, deep learning, intelligent control, neural networks**

I. Introduction

Rising global utilization of AI technology has led to questions regarding its sustainability [4]. With a rising percentage of global energy consumption [5] and an increasing carbon footprint [6][7], reports have suggested that the unregulated adoption of AI technology could become unsustainable in the near future [8]. A particular domain of concern is large vision models which can have high energy consumption during runtime [9] [10] [11]. This consumption is amplified when an AI system employs multiple vision models per-input. Such demands contribute to concerns about AI's sustainability and limit AI's accessibility due to financial cost [12]. In conjunction, a collaboration of models can extend task completion time, impacting the viability of multiple model systems in some applications.

The current approach to designing multi-vision-model systems is to have all models active for each input [13] [14] [15] [16]. Tasks that require complex vision systems may include models that are irrelevant to a significant proportion of inputs, examples being automated driving [17], robot-environment interaction such as Boston dynamics ”Spot” [18], GPT constructed systems [1][2], and intelligent multi-domain video surveillance [19]. Incorporating a precursor model into a system that prevents wasted model activation could reduce task completion time and energy usage. This could increase the time viability for certain tasks and the long-term sustainability of multi-vision-model systems.

In Figure 1 we see an example of unnecessary model activation. The input is irrelevant to two models in the system. However, altering a system to only activate relevant models can significantly reduce resource consumption. Figure 1 displays two ways we can reduce unnecessary resource consumption. Firstly through input-specific function activation and secondly, through image complexity-based model selection where larger models for ”tricky” inputs.

In this paper AI control refers to AI models instructing behavioral changes for any type of function. The predominant focus of AI control research pertains to control over *non-AI* functions. Examples include efficiently managing cloud resources [20], regulating voltage in power grid operations [21], and optimizing smart manufacturing systems [22]. Works on AI control over non-AI functions cannot be directly applied to scenarios involving AI control over AI functions due to significant variations in evaluation approaches and input processing.

We define AI models that instruct behavioral changes specifically for AI functions as AI-on-AI control. Previous literature for AI-on-AI control is limited, and research can be split into two main categories. The first are AI-based function selector's which build a multi-model AI system capable of executing a given task [1][2]. The second are AI controller's which

J. Burton-Barr, B. Fernando and D. Rajan are affiliated with the School of Computer Science and Engineering, Nanyang Technological University.

J. Burton-Barr and B. Fernando are affiliated with the Institute of High-Performance Computing, Agency for Science, Technology and Research, Singapore, and Centre for Frontier AI Research, Agency for Science, Technology and Research, Singapore.

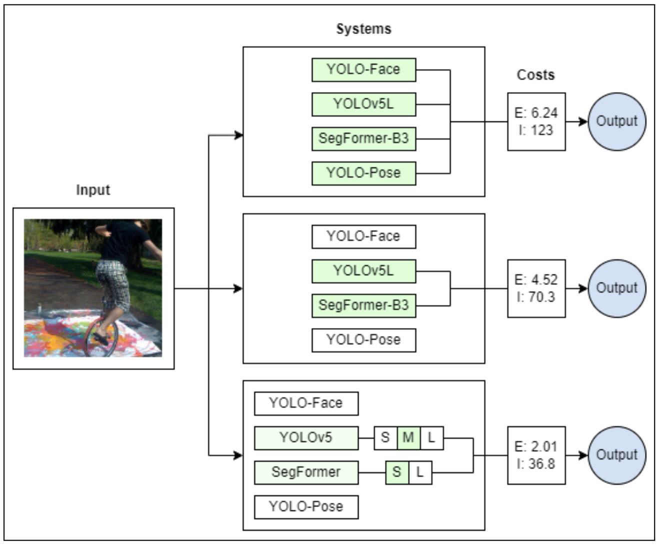

Fig. 1. Example of cost impact from unnecessary model activation. Under costs, ”E” stands for energy (in joules) and ”I” inference time.

receive an input and select a *single* best-fit model for that input to be processed by [3][23]. A literature gap is presented for scenarios involving input/model irrelevance in systems requiring co-active AI functions. Additionally, in [1][2][3][23], AI-on-AI control increased cost, leaving the application of AI-on-AI control towards cost-reduction and sustainability to be explored.

The framework we propose takes inspiration from works surrounding resource allocation in the human brain. The brain must perform well in a wide range of tasks with limited resources to allocate [24]. Core research believed that the frontal lobe central executive played a prominent role in resource distribution and input attention [25] in a modular human brain [26][27]. While the localization of such mechanisms is questioned by some research [28], support of organized resource distribution remains foundational [29][27]. The idea of a localized control unit that observes an input and then distributes input-related instructions can benefit AI systems. In particular, our framework seeks to bridge the gap for co-active multi-model systems, contributing insights to enhance resource utilization and address challenges related to AI system costs.

This study introduces an AI-on-AI control methodology tailored for co-active multi-vision-model systems. Our Supervised Image Classification Evaluative Controller (SICEC) employs image classification to preprocess inputs and make decisions regarding model activation within the system. Diverging from prevalent AI control methods, SICEC serves a dual purpose: overseeing AI models and enabling multi-function selection on a per-input basis. Function selection includes function activation/deactivation through a function controller (FC) and selection between same function-associated models with differing architectures through a secondary same functional controller (SFC). Our research aims to conceptualize and validate SICEC and explore its potential to enhance efficiency in multi-model AI systems. Instead of activating all models for each input in a given task, we aim to create an AI controller that understands the input and identifies necessary model activations. This reduces redundant processing from activating models irrelevant to the input. We aim to bridge the aforementioned gap in input-specific control for reduced resource consumption and make the following contributions:

- The novel application of image classification for activation control of co-active multi-model systems.
- Progression towards AI systems capable of adapting functionality on a per-input basis.
- Methodology for input-complexity evaluation enabling model size selection to reduce vision-system function costs.
- Discussion promoting the development of AI-on-AI control for resource efficient multi-model AI systems.

Note that SICEC is developed as an initial version of AI-on-AI control for future work to expand on. We are using SICEC to present the viability of image classification for cost-reductive AI system control. Our motivations are two-fold. Firstly, employing multiple AI models will usually increase a systems response time, better system management and removal of irrelevant input-model processing can reduce response time. Secondly, with the high-energy costs of AI and its sustainability being questioned, cost-reductive AI research can reduce both financial costs and the environmental impact of AI. If AI research takes a direction where the employment of multiple models for a problem increases, reducing environmental impact is particularly important.

II. Related Research

Rising reliance on power-heavy computational technology has motivated computational sustainability research. In 2014, [30] estimated the yearly energy consumption growth of three information and communication technology (ICT) categories. The growth of communication networks (10%), personal computers (5%), and data centers (4%) was higher than the 3% global average growth between 2007 and 2012. In 2015, researchers forecasted that communications technology could consume up to 51% of global electricity by 2030 [31]. These and similar works [32] [33] helped highlight a need for research on energy-efficient computing. This need is partially fulfilled by research directly focused on ICT efficiency, including but not limited to scheduling algorithms [34] [35] and low-power servers [36] for data centers, GPU power management [37] to increase computational hardware efficiency, and reducing redundant data transmission for internet of things [38].

Artificial intelligence is an additional category of computation with rising sustainability concerns. While discussion on the unsustainable nature of AI has become more common [39] [40], rarely does research directly focus on reducing AI energy consumption, instead focusing more on accuracy [41][42]. Training models with large parameters can incur substantial financial and environmental costs due to energy requirements [41][42]. While the energy cost per inference is relatively small, global consumption escalates as more devices make use of AI models [10]. In turn, there is a growing pressure for more sustainable AI technology and research.

Cost-reductive Computer Vision research has partially relieved the pressure. YOLO [43][44], EfficientNet [45][46], ShuffleNet [47], and MobileNet[48] focus on efficient architecture to reduce inference time. For example, YOLO architectures reduce cost through single-pass architecture, grid-based processing, and sharing of convolution layers. All of the aforementioned models network width and depth variants, which further reduces costs. While focused on inference time reduction, energy consumption is also reduced due to lower operations, parameters, and training times.

Efficient computer vision systems also appear in research. EVA^2 uses previous frames to predict the contents of the current frame and skip current frame processing if the content similarity is high [49]. Frame skipping reduces energy consumption and task completion time as later-stage models do not process frames with repetitive information. The authors in [50] alter image representation by feeding the frequency domain directly into the network as opposed to the standard CNN method of block-wise frequency decomposition to an expanded pixel representation and later stage re-transformation in-network processing. This avoids back-and-forth transformations during input processing and GPU/CPU switching, reducing inference times and, in turn, energy consumption. SICEC is closer related to [49] and [50] with a focus on an efficient system as opposed to efficient model architectures.

Within the field of AI controllers, a focus on cost-reductive research is present for AI control over non-AI systems. Authors [20], [21], [51] present the idea that an AI controller can learn to optimize the resource consumption of the system. Deep reinforcement learning (DRL) was applied by [51] for task scheduling in cloud computing platforms. The best-performing method, DRL with long short-term memory (LSTM), reduced CPU and RAM usage by up to 67% CPU and 72% respectively. The LSTM component provided the ability to store a set of previous states to better predict the consumption of future steps, which the DRL component utilized when creating an optimal scheduling policy. Our work follows the theme of resource reduction but explores situations where the AI controller interacts with a set of AI functions.

AI-on-AI control currently does not consider applications of AI control for cost-reduction of AI systems. Available works can be categorised as either *system construction*, where an upper-level AI model builds a system for a task, or *single model selection*, where the AI controller selects the best-fit model to process a specific input. HuggingGPT (H-GPT) [1] and Visual GPT (V-GPT) [2] are examples of *system construction* where an upper-level AI model selects a set of models, all of which are indiscriminately activated, for all inputs. H-GPT utilizes a user request and leverages models from HuggingFace, utilizing a model's written description as a basis for selection. V-GPT selects from a collection of visual foundation models and primarily differs from H-GPT by allowing multiple stages of user requests where the models employed may change accordingly. V-GPTs multi-stage approach to task completion means that irrelevant models can be removed based on the later stage user requests. H-GPT and V-GPT do not aim to understand the varying necessity of model activation depending on a specific input in a set. Our work seeks to create a controller that understands the relationship between input features and a model's function to reduce wasted model activation.

Works found on model activation specific to input features have aimed for *single model selection*. Such work pre-processes an input and selects the single ”best” model for that input from a set of models. For instance, in [3], a two-stage approach for input-specific model activation was developed. The first stage involved categorizing the perturbation type of an input and forwarding it to a relevant counter-perturbation network for second-stage processing. Similarly, [23] employed a delegation pre-processing phase to direct inputs to a single expert model. Both [3] and [23] use a pre-processor for an increased cost/accuracy trade-off. Such works do not aim to create a pre-processor capable of decision-making across multiple later-stage models nor seek to reduce costs for larger-scale AI systems.

In summary, we have highlighted:

- SICEC methodology is more closely related to cost-reductive methods that do not impact a model's architecture.
- While AI control over non-AI systems includes a cost-reductive focus, this methodology is limited in application to AI components.
- The methodological approach for H-GPT and V-GPT inherently differs from our research goals. Models within a constructed system uniformly process inputs during a user-defined task. Further, GPT usage increases the cost of task completion.
- Current works on AI systems with input-specific allocation only focus on single model allocation and how this can increase the accuracy of a system.

III. Proposed SICEC system

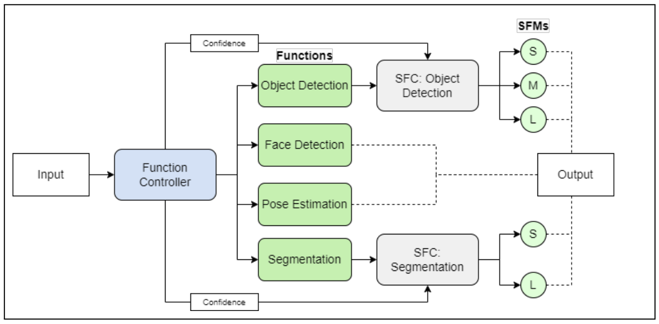

Fig. 2. SICEC overview.

Figure 2 displays the dynamics of a system that employs SICEC. *Functions* are unique AI-based behaviours linked to an AI model or models that the function controller can activate. The *Function Controller* is a single-label image classification model that pre-processes an input and activates relevant system functions. *Same Functional Models (SFM)* refer to a set of models that differ in architecture but share the same function. *Same Functional Controller (SFC)* refers to a method that decides which same functional model is most appropriate for the input. In this work, the SFC uses the function controller's *confidence* to select a same functional model. The *dotted lines* connected to the output represent that any function-related model may contribute to the system's output.

In Figure 2, the function controller, which has learnt specific features for each functions activation criteria during training, takes an image as input and outputs which functions are relevant to that image. The confidence of the function controller's decision is parsed to the SFC, which makes the same functional decisions, selecting weather small, medium, or large model is suitable for the function. Not all functions have a set of same functional models. In this case, activation of the function guarantees the use of a single associated model. The selected model of each function takes the image as input and outputs extracted information for further use. The following gives a formal overview of how SICEC functions:

Each system has an associated set of functions

$$M = \{m_1, m_2, m_3, ..., m_n\}. \tag{1}$$

Our system has four functions: Object Detection, Face Detection, Pose Detection, and Segmentation.

Each function *might* have a *set* of sub-functions

$$S(m_x)\{s_1, s_2, s_3, ..., s_i\}, \tag{2}$$

where i is the number of sub-functions that can vary for different functions within the system.

In our system, the object detection function has small (S), medium (M), and large (L) sub-functions, and the segmentation function only has small and large sub-functions.

There is a controller model that selects which functions m_i are appropriate for a given input x and also produces confidence score c

$$F(x) = (\{m_i\}, c). \tag{3}$$

Confidence score c is calculated by getting the maximum value of the softmax class confidence output

$$c = \text{argmax}\left(\frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}\right), \tag{4}$$

where z_i is the raw score associated with class i and N is the total number of classes.

If a function with associated sub-functions is selected, a separate function selects a sub-function. Our system uses a confidence score c produced by F

$$G(m_i, c) = S_j \in S(m_i), \tag{5}$$

where S_j is the selected sub-function.

The output of SICEC is the set of all activated functions without sub-functions, together with all the activated sub-functions. The function controller prevents the processing of inputs by irrelevant functions and the SFC reduces the employment of unnecessarily large function-related models. This, in turn, reduces the total inference time and energy consumption of the SICEC associated system.

A. Input Relevance Task

We aimed to design a task that exemplified multi-vision-model systems with cases of input-function irrelevance. Our task involves identifying animate objects and conditionally extracting information about humans within an input. Certain functions have interdependencies, reducing the decision space to six classes: **(1)** no activation, **(2)** object detection, **(3)** object detection and segmentation, **(4)** object detection, segmentation, and pose estimation, **(5)** object detection, segmentation, and face detection, and **(6)** object detection, segmentation, face detection, and pose estimation. The interdependency originates from function activation conditions outlined in Table I. For example, segmentation's activation condition will always be true if the activation conditions of face or pose detection are satisfied.

B. Function Controller

The *function controller* aims to understand each function's activation condition. These features are learned during training from a dataset that contains labels detailing modular relevance. The function controller takes an image as input, detects its features, and then outputs a set of functions relevant to the input. This version of SICEC's function controller uses EfficientNet [45] for single-label classification of inputs, with the objective of reducing inference time and energy consumption while keeping the SICEC's function controller cost relatively low.

C. Same Functional Models

SICEC can reduce the activation costs of functions by enabling the use of lower-cost models for less complex inputs. For example, a smaller model may be appropriate for an input with only a few easily distinguishable objects. In comparison, a larger model may be necessary for an input with many objects that are difficult to distinguish due to occlusion or other factors. In this paper, we aim to demonstrate a method for differentiation between functionally identical models as a foundation for further optimization in future research.

In Figure 2, same functional models (SFM) are sub-modular models that aim to produce identical outputs for the same input but differ in architecture. A same functional controller (SFC) is required to select an appropriate SFM. An SFC has the following properties:

1) Given some set of sub-functions (SFMs), our SFC should be able to choose a single sub-function most appropriate to the input.
2) This is a separate unit from the function controller and takes another step to decide between sub-functions.
3) Our SFC makes a decision between sub-functions using some already available information and without reprocessing the input.

Our SFC method inputs the function controller's output confidence and outputs a selected SFM. Lower function controller confidence leads to the selection of a more expensive model. This method incurs minimal cost as it does not process the image passed to SICEC. Instead, it has previously learned a set of confidence ranges that declare the appropriate SFM. The confidence ranges are based on a cost/benefit model distribution found by the following:

We take the classification confidence of 2500 inputs to the function controller, confidence calculated as shown in Equation 4. This is needed to compute a confidence distribution. We start by calculating the cost/benefit values for each SFM.

TABLE I
TASK SYSTEM MODELS.

		Activation	Inference		Energy
Function / Sub-function	Model	Condition	Time	GFLOPS	(Joules)
Object Detection S (OD)	YOLOv5s	Animate object present	15.5ms	16.5	0.41
Object Detection M	YOLOv5m	Animate object present	21.3ms	49.0	1.22
Object Detection L	YOLOv5l	Animate object present	35.5ms	109.1	2.72
Segmentation S (SEG)	SegFormer-B1	Person present	15.5ms	16.0	0.40
Segmentation L	SegFormer-B3	Person present	34.8ms	79.0	1.97
Face Detection (FD)	YOLOv5m-face	Face present	29.2ms	48.3	1.21
Pose Estimation (PE)	YOLOv5s-Pose	3+ people present	23.5ms	17.1	0.43
Function Controller (FC)	EfficientNet-B2	Each input	24.8ms	1.1	0.03

Given a set of SFMs, ($A_j^{\text{SFM}}, \forall j = \{1, \cdots n\}$), we obtain the normalised accuracy score for i-th SFM (A_i^{SFM}) as follows:

$$\hat{A}_i^{\text{SFM}} = \frac{A_i^{\text{SFM}}}{\sum_{j=1}^{n} A_j^{\text{SFM}}}, \tag{6}$$

where SFM_j is the reported accuracy in the original papers of respective SFM.

Given a set of SFMs, ($T_j^{\text{SFM}}, \forall j = \{1, \cdots n\}$), we obtain the reverse normalised inference time (so lower inference time is favoured) for i-th SFM (T_i^{SFM}) as follows:

$$\hat{T}_i^{\text{SFM}} = \frac{1}{n-1}\left(1 - \frac{T_i^{\text{SFM}}}{\sum_j^n T_j^{\text{SFM}}}\right), \tag{7}$$

where T_j^{SFM} is the inference time achieved on the RTX 2060m used for our experiments.

Following equation 7, we add the sets and re-normalise to get $\hat{C}^{\text{SFM}}$

$$\hat{C}^{\text{SFM}} = \frac{\hat{A}_i^{\text{SFM}} + w(\hat{T}_i^{\text{SFM}})}{\sum_{j=1}^{n}\left(\hat{A}_j^{\text{SFM}} + w(\hat{T}_j^{\text{SFM}})\right)}. \tag{8}$$

where w is the weight for prioritization of T_i^{SFM} and $\sum$ is the summation of sets $\hat{A}^{\text{SFM}}$ and $w(\hat{T}^{\text{SFM}})$ together.

In Equation 8 $\hat{T}^{\text{SFM}}$ is multiplied by w, allowing for prioritisation of cost over benefit, benefit being the reduction in resource consumption. If $w < 1$, then benefit will be prioritised over cost; $w > 1$ prioritises cost over benefit. In our system design we prioritise cost and set the value of w to 2.

Initially segmentation and object detection consisted of S and L SFM variants, an M variation is appended to object detection to test the impact of increasing available SFMs. Following Equation 8 the confidence dataset list is sorted high to low. For object detection the normalised values gained in Equation 8 are used to proportionally split the list into S, M, and L confidence value sets (e.g., for an S value of 0.45 would receive the first 45% of the list). For S, we assign the upper range to 100 and the lower range to the lowest value in its associated confidence set. We take the S set's lowest value and the L set's highest value for the M range. Finally, the L confidence assignment range is the highest value of the L set to 0. When our function controller makes a decision, the decision confidence is sent to the SFC and checked against the confidence ranges for S, M, and L for the object detection SFM's or S and L for the segmentation SFM's. Equation 9 shows SFM selection for object detection, with bigger models being assigned images with less certainty.

$$S_j = \begin{cases} Small & \text{if } 100 \geq x \geq min(s_c) \\ Medium & \text{if } min(s_c) > x \geq min(m_c) \\ Large & \text{if } min(m_c) > x \geq 0 \end{cases} \tag{9}$$

where c is the same functional model associated confidence range, S_j is the SFC decision 1 to j, x is the function controller confidence.

IV. DATASET AND METRICS

In this section, we describe datasets created for the function controller and then additional metrics for inference time and accuracy performance. Following this, we outline the SFC datasets and additional SFC metrics. Following, a sub-section is dedicated to energy analysis, including energy metrics, and finally, a sub-section on function controller implementation.

A. Function Controller Datasets

Auto-Annotation Refined: A time-practical approach of auto-annotating a set of images was used. A set of models would be run on each image and then checked for the activation conditions for the input relevance task. Each image would then be classified with the correct decision perceived by auto-annotation. Figure 3 shows the auto-annotation process for the input relevance task.

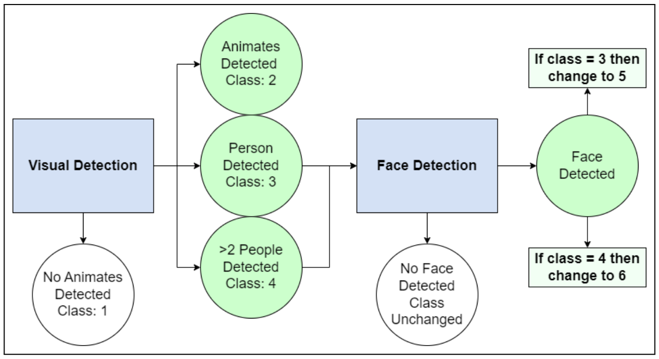
Visual Detection
No Animates Detected Class: 1
Animates Detected Class: 2
Person Detected Class: 3
>2 People Detected Class: 4
Face Detection
No Face Detected Class Unchanged
Face Detected
If class = 3 then change to 5
If class = 4 then change to 6

Fig. 3. Auto-Annotation process for the input relevance task dataset.

Figure 3 outlines the auto-annotation process for creating the input relevance task dataset. Each image is parsed to the auto-annotation in a multi-step process. First, the input is parsed to an object detection; if no animates are detected, the input is assigned class 1. Depending on if people are present in the input and the quantity of people, the input is assigned class 2, 3, or 4. If the input has been assigned 3 or 4, it is parsed to the face detection model, and if faces are detected, the input is assigned a new classification accordingly.

The original output of auto-annotation is a 5958-piece subset of COCO [52] used as the primary function controller training dataset. The dataset is split into 85% training and 15% test images. Initial experiments showed that relying on auto-annotation significantly decreased function controller performance due to labeling errors. A script was produced that allows human-assisted auto-annotation where a person can remove incorrect annotations, creating a refined auto-annotation. This reduced the dataset size to 4482.

Function Controller Validation: A refined auto-annotated ImageNet (Deng et al., 2009) validation set was also created with 1745 images. These were split equally across the six classes. Due to the inconsistency of faces between classes previously discovered, we removed heavily occluded faces from the validation set.

B. Function Controller Metrics

Standard accuracy metrics such as *mean average precision* (mAP) and recall were insufficient to measure SICEC's performance. Additional metrics are employed to measure the performance (cost and accuracy) of the SICEC's components (function controller and SFC).

System Costs: An assumption is made that the non-SICEC system uses all available models for each input as without a function controller, this would likely be standard operation. The same cost equations are used for total inference, GFLOP, and energy cost. Also, if SFC is excluded from the calculation, the highest cost SFM is assumed.

The total cost of the proposed SICEC system is given by

$$SICEC\ TC = \frac{1}{a}\sum_{i=1}^{a}\left(\sum_{j=0}^{n} MC_{ij} + FC + \sum_{k=0}^{m} SFM_{ik}\right), \tag{10}$$

where i is the i-th image in image set $1\ to\ a$, MC_{ij} is the model cost for each non-SFC selected function $0\ to\ n$, SFM_{ik} is the model cost for each SFC selected model $0\ to\ m$, and FC is the function controller cost.

Without SICEC, for each inference the cost for each function within the system is used

$$Standard\ TC = \frac{1}{m}\sum_{i=1}^{m}\left(\sum_{j=1}^{n} MC_{ij}\right), \tag{11}$$

where i is the i-th image in image set $1\ to\ m$ and j is the maximum model cost for each function $1\ to\ n$.

Closeness: SICEC can be partially correct which is not considered by standard Mean Average Precision (mAP). We calculate closeness for each input through Jaccard's similarity, which better represents SICECs accuracy

$$Closeness = \frac{1}{n}\sum_{i=1}^{n}\frac{|A_i \cap B_i|}{|A_i \cup B_i|}, \tag{12}$$

where i is the i-th image in the image set $1\ to\ n$, A is a set of function controller selected functions for the i-th image, and B is the correct function activations for the i-th image.

Note that Equation 12 tells us the closeness of SICEC, including correct decisions. A closeness value higher than validation accuracy implies the presence of partially correct decisions. However, Equation 12 alone does not give a good representation of incorrect cases. *Independent closeness* uses the same equation as Equation 12 but ignores entirely correct cases, isolating the closeness of erroneous decisions. A high independent closeness indicates that some input-relevant functions are still activated when SICEC fails to make a correct decision.

Correct Model Activation: Similar to the closeness equation, however it only considers correct functions activated. This means that given a set of functions the function controller has activated, we only consider how many functions match the true decision. Correct model activation better represents the task accuracy compared to our closeness metrics.

$$Correct\ Model\ Activation = \frac{1}{n}\sum_{i=1}^{n}\frac{|A_i \cap B_i|}{|B_i|}. \tag{13}$$

C. SFC Metrics

SFCs are challenging to define as correct/incorrect. Determining when one model of the same function prioritises over another is an ambiguous concept. For results on SFC cost-reduction, we can separate the SFC section of Equation 10 and make a comparison to a non-SFC SICEC.

However, this does not fully specify the SFC's performance. Assignment validation metrics can give more information on this. We compared the mAP of SFC-assigned models to the expected mAP or accuracy of that SML distribution. For the object detection SFMs, we get the S, M, and L SFM distribution on a 2500-piece COCO validation subset. Then, we run each model across the subset to get their mAP, which we can use to calculate the expected mAP.

$$Expected\ Accuracy = \left\{\frac{1}{m}\sum_{i=1}^{m}\frac{1}{n}\sum_{j=1}^{n}(IAM_{ij} * TMA_{ij})\right\}, \tag{14}$$

where i is the i-th SFM in SFM set $1\ to\ m$, j is the j-th image in image set $1\ to\ n$, IAM is the SFC assigned model for the j-th image, TMA is the original paper reported mAP of that model.

We then run each model on its SFC-assigned images and get the mAP for each model and the overall mAP for the subset

$$SFM\ Assigned = \left\{\frac{1}{A}\sum_{i=1}^{A} IAM_{ji} \mid j \in [s, m, l]\right\}, \tag{15}$$

where i is the i-th image in image set $1\ to\ A$ and IAM_j is the AP achieved on the i-th image by the j-th SFM

The same process is done for the segmentation SFMs except on the 2000 piece ADE validation set and now only an S and L model ($\in [s, l]$ opposed to $\in [s, m, l]$) is used. Also, Mean Intersection Over Union (mIoU) is used over mAP. Classes not present in the S or L set are removed from accuracy calculation as they would receive a score of 0, biasing the mIoU.

For both the object detection and segmentation SFMs, we perform *cross-testing*. Once the SFMs have an associated image set, we test each SFM on the other sets. The performance of an SFM on the other image sets can support if the images have been logically assigned. For example, if the S model receives a lower mAP or mIoU on the M and L set, then we have shown that the images in the M and L set are more difficult for the S model to process, supporting logical assignment. However, if the S model performed better on the M or L set or roughly the same, this would imply images have not been logically assigned to an SFM.

D. Measuring energy

Model energy usage is inconsistent across hardware platforms. However, the FLOPS of a model is consistent, with a significant reduction in operations implying less processing. To calculate energy consumption we acquire the flops per watt performance of the RTX 2060m:

$$\text{Performance per Watt} = \frac{\text{Max FLOPS/s}}{\text{Max Wattage/s}} \tag{16}$$

Models employed for IRT and SICEC use FP32, at which the theoretical limit of the RTX 2060m is 4.608 TFLOPS/s. The maximum power consumption of the RTX 2060m is 115w/s, using Equation 16 our GPU performs approximately 40.7 GFLOPS per watt. We use this to get the energy usage, in joules (J), for each function and SFM in the SICEC system:

$$J = \frac{\text{Model FLOPS}}{\text{Performance per Watt}} \tag{17}$$

For calculating SICEC and standard system energy metrics, we use Equation 10 replacing inference with energy. The energy cost per inference includes each model activated, the function controller, and SFC. Further, like inference cost calculation, the SFC portion of Equation 10 is ignored for non-SFC SICEC. Equation 11 is still used for the non-SICEC system comparison and the highest cost models are assumed.

E. Function Controller Implementation

The function controller of SICEC uses EfficientNet (EN). Three versions (EN B0, B2, and B4) of the function controller are trained on the refined dataset to test the impact of resolution, width, and depth on function controller classification. Increasing resolution and image processing can help the function controller pick up more minor details but come at a higher cost. EN was primarily chosen to keep the function controller costs relatively low compared to the total cost of the models used for the task.

The function controller was trained to produce a single label encompassing multiple decisions. The total number of classes is six. The animates for object detection function activation are limited to cats, horses, sheep, birds, dogs, and people. The training conditions included a batch size of 8, 20 epochs, a learning rate of 0.00025, weight decay of 0.8 every 3 epochs, and momentum of 0.9. The input size for B0 was 224, B2 was 260, and B5 was 456. The loss function used was cross-entropy, and the optimizer employed was stochastic gradient descent.

We use an RTX 2060m (6 GB) for hardware with an Intel i7-9750H CPU for training and testing. Inference times and energy use recordings are done for inputs without the use of batching. We feed the images to the function controller and the system functions/sub-functions individually. We would likely see a lower cost per image if batching were used.

V. Experimental Results

The results section is organised by first discussing functional controller results and then SFC results without energy consumption analysis. Following the analysis of the SFC, we compare the energy performance of both the function controller and SFC together. Finally, key limitations and implications for our methods are discussed. EfficentNet-B2 achieved a *correct model activation* close to B4 and offered a good middle-ground for inference cost (Table II). Given the focus on SICEC's impact on cost-reduction, our results section mainly discusses an EfficientNet-B2 function controller.

A. Function Controller Accuracy, Closeness, and Correctness

Table II presents the accuracy, closeness, and correctness of SICEC function controller variations. As expected, increasing the width and depth of EfficientNet improves decision accuracy. Increased network processing at higher image resolution enables the detection of finer class-specific differences. Test accuracy is 73.57% for B0, 76.37% for B2, and 81.45% for B4, suggesting that B4 is significantly more competent at model activation compared to B0 and B2. ImageNet validation repeats the trend; however, EfficientNet B0 (73.79%), B2 (78.15%), and B4 (82.34%) all show an increase in validation accuracy. This increase is likely due to reduced facial ambiguity from removing heavily occluded faces due to conflicts in class placement. Examples of correctly identified cases are displayed in Figure 5.

The performance gap closes between B2 and B4 for our introduced metrics. Closeness shows that the model activation and performance of B2 (87.04%) and B4 (89.30%) are better than test/validation accuracy suggests. Independent closeness shows that given a misclassification B2 (45.52%) and B4 (45.56%) still activated a proportion of the correct models. For correct model activation, the performance difference between B2 and B4 is less drastic than test/validation accuracy suggests, with B2 achieving 88.24% and B4 achieving 90.25%. Despite misclassification, our function controller still partially identifies model-relevant features. With a reduced performance gap and significantly lower inference time, EN-B2 is used for the function controller in the remaining results.

B. Function Controller Confusion Analysis

Figure 4 shows the confusion metrics for our function controller. We see that the function controller particularly struggles with class 4, often failing to activate the pose function. Figure 5 visualises how the function controller has

TABLE II
SICEC Function Controller Results

Model	Test Accuracy	Validation Accuracy	Closeness	Independent Closeness	Correct Model Activation	Inference Time
EN-B0	73.57%	73.79%	84.03%	42.98%	85.20%	18.8ms
EN-B2	76.37%	78.15%	87.04%	45.52%	88.24%	24.8ms
EN-B4	81.45%	82.34%	89.30%	45.56%	90.25%	34.2ms

difficulties activating the pose function where three to five people are present or bodies overlap. Class 3 is also commonly misclassified with class 1 or 2 which mostly occurs in cases of high object occlusion, potentially caused by inadequate representation due to small dataset size. The incorrect rows of Figure 5 further highlight that the function controllers difficulties with occlusion was a significant source of inaccuracy.

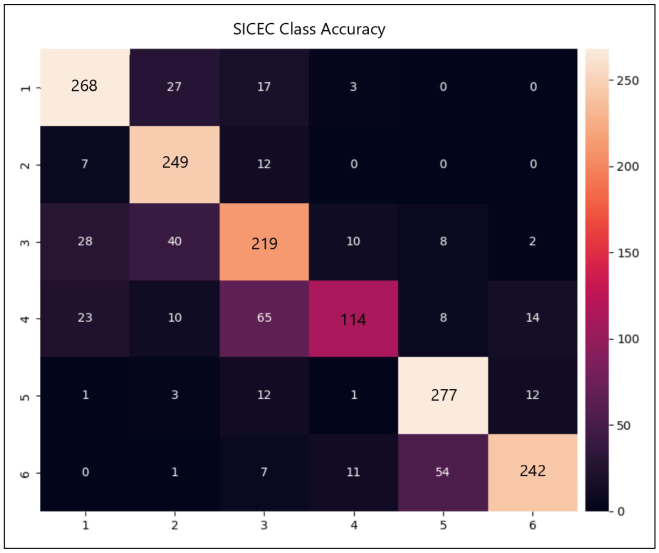

Fig. 4. EN-B2 confusion matrix for validation accuracy.

The correct identification of classes is shown in the first two rows of Figure 5. In ”object detection”, we see correct classification of the animate class and a case in ”none” where a non-specified animate has received the correct classification. The function controller can also pick up relatively small faces in classes 5 and 6. While the activation of pose estimation is unreliable for smaller groups of people, for larger groups, the function controller is reasonably competent. For classes 1 and 2 Figure 4 shows that SICEC is unlikely to activate functions 4, 5, and 6, reducing the inference and energy penalty for misclassification. Misclassification of 3 and 4 also more commonly leads to under-selection oppose to over-selection of functions. The closeness of function controller's classification generally increases with abundance of relevant features. For example, in Figure 4 classes 4 and 6 are unlikely to be misclassified as 1 or 2 as multiple people are present. Figure 5 exemplifies this in the class 6 incorrect section. Where pose has failed to be activated, an abundance of features relevant to face detection (class 5) remain present.

A problem we observed originates from feature similarity between classes and function controllers single label design. Features of each class overlap, for example, class 6 shares features with all previous classes and class 3 and 4's relevant features are very similar. We term this as *feature confusion*, where high feature similarity between classes leads to misclassification. For example, The similarity of class 4 to 3 could contribute to the unreliable class 4 classification for smaller groups. Potentially, where less people are present, the confidence in class 4 lessens, while the confidence rises for class 3 due to abundance of relevant features (Figure 5 class 4 incorrect). For class 5, the confidence could lessen due to facial occlusion and the class 3 confidence overtakes (Figure 5 class 5 incorrect).

C. SFC Results

Table III shows that the SFC methodology is partially successful. For our object detection SFMs, we see an appropriate trend of S and M accuracy increasing with a decrease in L accuracy. This suggests that complex images are being assigned to the L model. The SFC maintains the average accuracy across the three models, which is expected given the decrease in L accuracy. The S accuracy increase and L accuracy decrease are repeated for segmentation.

TABLE III
Assignment vs Distribution. **Standard** refers to the S, M, or L models normal performance on the validation set before being assigned a set of images.

Condition	S	M	L	AVG
Standard OD	0.384	0.468	0.505	0.452
Assigned OD	0.410	0.497	0.451	0.453
Standard SEG	0.529	N/A	0.580	0.555
Assigned SEG	0.548	N/A	0.554	0.551

Tables IV and V show the results of SFM cross-testing, which refers to the testing of each SFM on other SFM-assigned image sets. Table IV shows S and L cross-testing results reinforce the claim of logical image assignment. On the L set, YOLOv5s achieved an mAP of 33.8, 7.2 below the mAP on its own set. On the S set YOLOv5l, achieved an mAP of 52.4, 7.3 greater than the L-assigned image set. The same trend is seen for segmentation results in Table V.

TABLE IV
SFC Cross-Testing for Object Detection. S, M, and L Sets are the SFC-assigned image sets that the models are cross-tested on.

Model	S Set	M Set	L Set
S	0.410	0.415	0.338
M	0.489	0.497	0.417
L	0.524	0.535	0.451

While the S and L assignment has shown success, creating a middle-ground for intermediate models has been unsuccessful.

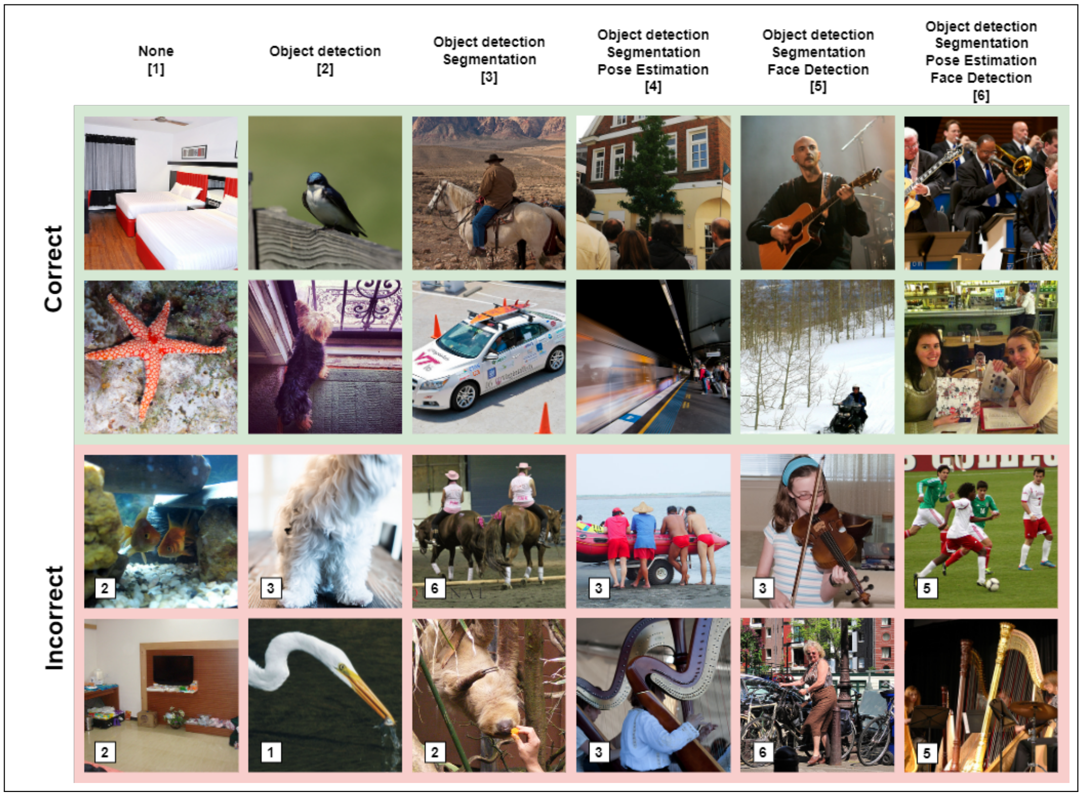

Fig. 5. SICEC EN-B2 correct and incorrect decision examples. Incorrect decisions are placed under their correct classes with a number tag that shows the incorrect SICEC classification. The two "animates" in [1] are not animates EN-B2 was trained to detect.

YOLOv5s achieves a 0.005 higher mAP on the M set than the S set, and YOLOv5l achieves 1.1 higher mAP on the M set than on the S set. This implies that many images assigned to set M are better suited to set S or vice versa. Such problems could arise from the function controller confidence output only being loosely connected to the SFM models. While image complexity (number of objects, small object size, occlusion, and noise) will impact vision models' performance, the extent and cause of impact varies between models. In extension, the EfficientNet model can have a high confidence unrepresentative of complexity if a relevant object takes up significant image space or multiple occurrences of an object support a classification. Same functional methodology has reduced the use of larger SFMs with evidence of some logical image assignment. However, results suggest SFC methodology and how image complexity is represented require additional attention.

TABLE V
SFC Cross Testing for Segmentation.

Model	S Set	L Set
S	0.548	0.508
L	0.554	0.608

D. Inference Cost Analysis

SICEC showed large reductions in our AI system costs. A visual summary of our cost results can be seen in Table VI. Without the use of SFCs, an inference time reduction of 35.0% was found using EN-B2 (80ms vs 123ms) per inference. This increased to 48.9% (62.8ms vs 123mms) with SFC. Without a SFC, the function controller method accounts for 31% of the inference time and 39.5% when the SFC is implemented.

TABLE VI
Cost Analysis of SICEC with and without SFC implementation. The **standard** system does not employ the Function Controller (FC) or SFC.

	Method	AVG
Method	Infer (ms)	(ms)
Standard	0	123.0
FC w/o SFC	24.8	80.0
FC w SFC	24.8	62.8

We can compare the performance of SICEC to a similar system built by H-GPT for the input relevance task. Note that H-GPT had no models available for face detection or a suitable substitute model. H-GPT employed the following models on an unknown GPU; brackets show the inference times returned by H-GPT: detr-resnet-101 for visual detection (200ms), detr-resnet-50-panoptic for image segmentation (500ms), and Openpose-control for pose estimation (200ms).

Figure 6 shows the results of SICEC using the H-GPT selected models. SICEC achieves an average inference of 469.8ms, 47.8% lower than H-GPT's 900ms. If we take the average inference time of the H-GPT models, we can employ an artificial face detection (A-FD) model with an inference time of 300ms. In this scenario, SICEC achieves an average inference time of 551.8ms, 54% lower than H-GPT's 1200ms. The lower relative cost of the function controller

(5.3% and 4.5% vs 31%) contributes to the increased inference reduction. This highlights that SICEC methodology becomes more beneficial when the relative cost of SICEC is smaller.

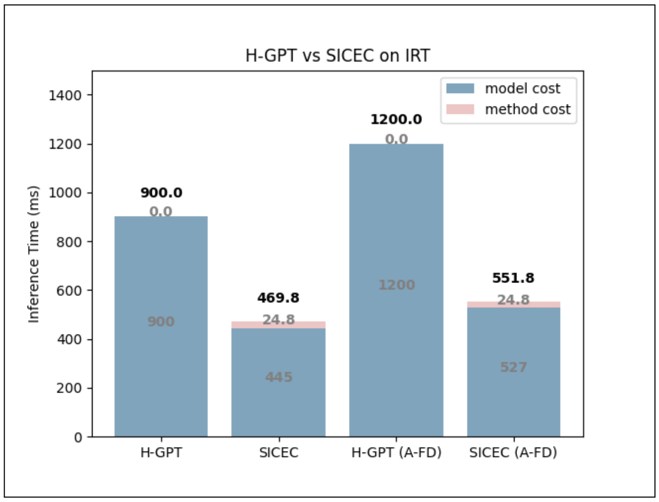

Fig. 6. Comparing H-GPT selected models with and without the implementation of non-SFC SICEC.

An additional comparison with V-GPT was not possible as only 2/4 input relevance task functions were available in V-GPTs visual foundations models. Further, when prompting *V-GPT* and *H-GPT* to attempt input-specific co-active multi-model control across a set of images, both responded that the function was not within their capabilities. SICEC separates itself from these GPT models with the ability to activate/deactivate models on a per-input basis. Further, GPT systems do not contain SFMs; SICEC was designed to utilise SFMs and only use larger models when necessary.

E. Energy Cost Analysis:

Assuming the relationship is consistent across hardware, then SICEC's reduced GFLOPs (Figure 7) infers a reduction of energy for a SICEC system across different hardware platforms. For our implementation, the reduction in GFLOPs was 47.8% without SFC and 67.8% with SFC. Table VII shows the reduction in energy usage was 43.7% (3.57 vs 6.34) and 66.4% (2.13 vs 6.34), respectively.

Table VII breaks down the average energy usage of the input relevance task models, and Table VIII shows the SFC's impact. The contributions of the SFC are seen in YOLOv5, which uses 0.83 joules per inference, a reduction of 54.1% compared to SICEC without SFC (1.81). The SegFormer SFMs used 0.61 joules per inference, resulting in a reduction of 43.0% compared to SICEC without SFC (1.07). It is implied that the SFC reduces SICEC's energy use by a further 1.44 joules per inference; this amounts to 34.0% of the total energy (4.23) saved by SICEC, 2.79 joules (66.0%) being saved by the function controller.

The function controller row in Table VII shows the relationship between input-model relevance and energy reduction. Models with lower general input relevance experience a greater energy reduction during task run-time. For the object detector function (highest input-relevance), a 33.7% consumption decrease and a 72.1% decrease for the pose estimation function (lowest input relevance) were found. This disparity shows the function controller's ability to reduce energy consumption is greater for functions with lower input relevance.

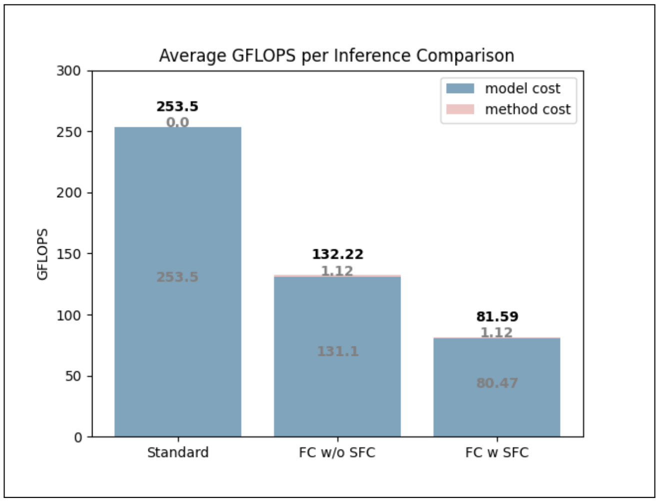

Fig. 7. SICEC's reduction of average GFLOPS per inference.

Figure 8 shows the contribution of each function towards each classes total energy consumption. Both object detection (OD) and segmentation (SEG) use the average energy consumption of the SFC selected model. The more functions activated the more the potential energy consumption of a single input varies, this generally depends on complexity and quantity of function relevant features in an image. Large standard deviations arising from activation of SFM related functions result from different model sizes. A higher frequency of perceived high complexity images would shift the consumption of SFM related function towards the upper limit of the standard deviation, increasing the SICEC system cost. The final bar in Figure 8 represents the standard non-SICEC system. We see an increment in segmentation and object detection energy cost, however, the energy variation per input decreases due to the removal of SFMs.

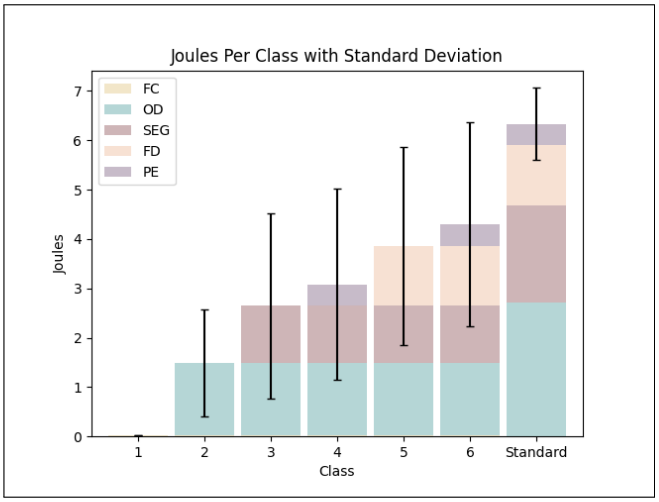

Fig. 8. Joules per function controller class.

TABLE VII
Function Energy Cost (J). Rows (standard, FC, and FC with SFC) show the average energy consumption per image.

Method	Object Detection	Segmentation	Face Detection	Pose Detection	Function Controller	Total
Standard	2.73	1.97	1.21	0.43	N/A	6.34
FC	1.81	1.07	0.54	0.12	0.03	3.57
FC with SFC	0.83	0.61	0.54	0.12	0.03	2.13

TABLE VIII
Extended SICEC cost-reduction from same functional control. ”Reduction” is compared to the functional controller without the SFC.

Method	S	M	L	Total	Reduction
Object Detection	0.15	0.14	0.54	0.83	54.14%
Segmentation	0.12	N/A	0.49	0.61	43.00%

With a total energy reduction of 66.4%, SICEC can significantly reduce energy costs. Different SICEC systems will contain different functions and associated models or could be linked to alternative tasks that may increase or decrease energy reduction. For the input relevance task, if we had used more computationally expensive models (E.g., those employed by H-GPT), it would have increased the energy saved. This would also occur for greater computational differences between small and large SFMs. A task with a lower activation frequency of functions could increase cost-reduction; however, tasks with a high activation frequency would decrease cost-reduction. This relationship between task requirements and system function costs is essential when deciding if implementing SICEC is worthwhile.

F. Comparison to other works

Limited comparative evaluation is available for SICEC. Previous work surrounding AI systems with multiple models [1] [2] [3] do not focus on cost reduction and sustainability. Therefore, performance indicators relevant to the goals of SICEC do not exist. Available work relatable to cost reduction and sustainability in AI focuses on single model applications. Previously mentioned, such work focuses on architectural changes to reduce model costs [45][48] or evading unnecessary processing through external input evaluation or adjustment [49][50]. The latter is technically comparable to SICEC; however, our methods overhead cost is justified by the savings from a set of models. In single model scenarios the function controller cost of SICEC could restrict cost reduction (energy consumption and inference time) of the application.

VI. Limitations

A. Function Controller

The discussion on Table II and Figure 4 highlights the key drawback of SICEC, its dataset dependency, and the issue of creating a clearly defined and task-representative dataset. Features can easily become ambiguous, and feature confusion occurs within training due to overlap between classes or difficulties defining the class in the dataset. For testing, it is challenging to create a test set that fairly represents how SICEC has learnt to control its functions. This impacts the ability for SICEC to correctly associate features with a function and the ability to accurately evaluate SICECs performance.

B. EfficientNet as a Function Controller

EfficientNet was selected due to being a lightweight classifier that would minimise the relative cost of the function controller. As shown in Table VI and Figure 8 our function controllers energy consumption was relatively low compared to other models in the IRT system. However, an inference time of 24.8ms contributed to 31.0% of the non-SFC SICEC and 39.5% of the SFC SICEC average inference time (Table VI). It is important to keep the relative cost of the function controller low to prevent unnecessary limitations on the system. In our case, the inference time of EN-B2 might contribute to a higher than necessary bottleneck on system response time.

C. Same Functional Controller

SFC and SFMs were novel topics to add an extra dimension of cost-reductive AI-on-AI control in SICEC. Given a poor object detection M performance, refinement of SFC methodology is needed to increase the logicality and reliability of assignment. Our use of function controller confidence is limited in predicting the SFMs performance on an input. Methods might improve if the definition of image complexity better discriminates between SFMs and more accurately captures model-input performance. Other SFC avenues exist which we chose not to explore in this work, fractal complexity evaluation [53] or tensor regression complexity classification being two examples.

VII. Implications

SICEC's impact on multi-AI systems will depend on various factors. System model sixes, long system run times, mass-distributed systems, real-time requirements, system financial budgets, and input-model relevance ratio can increase the appeal and justify the manual cost of implementing SICEC. Systems that meet the criteria can benefit from lower-resource computing. In application, this could reduce environmental impact and create more sustainable AI systems.

Implementing SFMs and a suitable SFC can independently impact the energy and inference costs of systems using AI models. SFCs can allow for the application of larger AI models without sole reliance on those models. Our findings showed that a significant proportion of SICECs cost-reductions resulted from SFCs and the availability of SFMs. SFCs can also reduce the burden of expertise and heavy cost-benefit analysis when making system-model selections.

With further development, SICEC can be versatile in its application to the AI industry. In application, SICEC also has incentives beyond sustainable AI. Cloud computing could benefit by having an intermediate SICEC processor that directs an input to relevant computer vision cloud services, reducing financial costs. For available GPT systems that assign models to a user-specified task, SICEC methodology could reduce computational waste for model-irrelevant inputs. This could make the cost of using GPT systems for AI tasks more viable. In robotics, we may see the implementation of multiple models to achieve some interaction with the visual environment. However, SICEC can advise these models and remove irrelevant model-processing leading to quicker task performance.

VIII. Conclusion

We have shown that SICEC can significantly reduce system costs while maintaining a high percentage of correct model activation. We have also displayed a working demonstration of logical SFM selection using a zero-cost SFC technique. With significant reductions in energy and inference costs, we showed the potential of SICEC to manage multi-vision-model AI systems and increase system efficiency. Our work has contributed to sustainable and low-resource AI through the novel application of image classification and input-complexity analysis for function activation and function-model selection. In turn, we have also provided foundation for further research on AI-on-AI control for resource-efficient AI systems.

Future work should focus on two limitations of our function controller. The first is feature confusion caused by ambiguity between classes in the dataset. The second is the manual efforts required for creating SICEC's dataset and difficulties for creators in handling class-feature overlap. Both limitations can be overcome through refinement of the dataset creation process and improving the distinctness of classes for the function controller. Future SFC methodology should explore comprehensive identification of image complexity and its relationship with model performance. This could increase both the logicality of SFM assignment and the appeal of SFC implementation in cost-reductive AI systems.

Acknowledgments

This research/project is supported by the National Research Foundation, Singapore, under its NRF Fellowship (Award NRF-NRFF14-2022-0001). Any opinions, findings conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore. This research is also supported by funding allocation to B.F. by the Agency for Science, Technology, and Research (A*STAR) under its SERC Central Research Fund (CRF), as well as its Centre for Frontier AI Research (CFAR). The authors would like to thank James Pallister and James Leetch for their assistance in preparing this paper for publication.

References

[1] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv preprint arXiv:2303.17580*, 2023.

[2] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023.

[3] Pratyush Maini, Xinyun Chen, Bo Li, and Dawn Song. Perturbation type categorization for multiple adversarial perturbation robustness. In *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022.

[4] Scott Robbins and Aimee van Wynsberghe. Our new artificial intelligence infrastructure: becoming locked into an unsustainable future. *Sustainability*, 14(8):4829, 2022.

[5] Rising global ai enery. https://news.climate.columbia.edu/2023/06/09/ais-growing-carbon-footprint/. Accessed: 13-09-2023.

[6] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

[7] Elizabeth Gibney et al. How to shrink ai's ballooning carbon footprint. *Nature*, 607(7920):648–648, 2022.

[8] Sophia Falk and Aimee van Wynsberghe. Challenging ai for sustainability: what ought it mean? *AI and Ethics*, pages 1–11, 2023.

[9] Radosvet Desislavov, Fernando Martínez-Plumed, and José Hernández-Orallo. Trends in ai inference energy consumption: Beyond the performance-vs-parameter laws of deep learning. *Sustainable Computing: Informatics and Systems*, 38:100857, 2023.

[10] Radosvet Desislavov, Fernando Martínez-Plumed, and José Hernández-Orallo. Compute and energy consumption trends in deep learning inference. *arXiv preprint arXiv:2109.05472*, 2021.

[11] Da Li, Xinbo Chen, Michela Becchi, and Ziliang Zong. Evaluating the energy efficiency of deep convolutional neural networks on cpus and gpus. In *2016 IEEE international conferences on big data and cloud computing (BDCloud), social computing and networking (SocialCom), sustainable computing and communications (SustainCom)(BDCloud-SocialCom-SustainCom)*, pages 477–484. IEEE, 2016.

[12] Abhishek Gupta, Camylle Lanteigne, and Sara Kingsley. Secure: A social and environmental certificate for ai systems. *arXiv preprint arXiv:2006.06217*, 2020.

[13] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 564–571, 2013.

[14] Fangting Xia, Peng Wang, Xianjie Chen, and Alan L Yuille. Joint multi-person pose estimation and semantic part segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6769–6778, 2017.

[15] Fábio Perez, Sandra Avila, and Eduardo Valle. Solo or ensemble? choosing a cnn architecture for melanoma classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[16] Ruoyu Fang and Cheng Cai. Computer vision based obstacle detection and target tracking for autonomous vehicles. In *MATEC Web of Conferences*, volume 336, page 07004. EDP Sciences, 2021.

[17] Sorin Grigorescu, Bogdan Trasnea, Tiberiu Cocias, and Gigel Macesanu. A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386, 2020.

[18] Spot by boston dynamics. https://www.bostondynamics.com/products/spot. Accessed: 05-09-2023.

[19] G Sreenu and Saleem Durai. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *Journal of Big Data*, 6(1):1–27, 2019.

[20] Muhammad Wajahat. *Cost-Efficient Dynamic Management of Cloud Resources Via Supervised Learning*. PhD thesis, State University of New York at Stony Brook, 2020.

[21] Jiajun Duan, Di Shi, Ruisheng Diao, Haifeng Li, Zhiwei Wang, Bei Zhang, Desong Bian, and Zhehan Yi. Deep-reinforcement-learning-based autonomous voltage control for power grid operations. *IEEE Transactions on Power Systems*, 35(1):814–817, 2019.

[22] Thanasis Kotsiopoulos, Panagiotis Sarigiannidis, Dimosthenis Ioannidis, and Dimitrios Tzovaras. Machine learning and deep learning in smart manufacturing: The smart grid paradigm. *Computer Science Review*, 40:100341, 2021.

[23] Yikang Zhang, Zhuo Chen, and Zhao Zhong. Collaboration of experts: Achieving 80% top-1 accuracy on imagenet with 100m flops. *arXiv preprint arXiv:2107.03815*, 2021.

[24] Dale Zhou, Christopher W Lynn, Zaixu Cui, Rastko Ciric, Graham L Baum, Tyler M Moore, David R Roalf, John A Detre, Ruben C Gur, Raquel E Gur, et al. Efficient coding in the economics of human brain connectomics. *Network Neuroscience*, 6(1):234–274, 2022.

[25] John N Towse and Carmel MT Houston-Price. Reflections on the concept of the central executive. *Working memory in perspective*, pages 260–280, 2002.

[26] Isabelle Brocas and Juan D Carrillo. The brain as a hierarchical organization. *American Economic Review*, 98(4):1312–1346, 2008.

[27] Margaret C Jackson, Helen M Morgan, Kimron L Shapiro, Harald Mohr, and David EJ Linden. Strategic resource allocation in the human brain supports cognitive coordination of object and spatial working memory. *Human Brain Mapping*, 32(8):1330–1348, 2011.

[28] Robert H Logie. Retiring the central executive. *Quarterly Journal of Experimental Psychology*, 69(10):2093–2109, 2016.

[29] Michael D Fox, Abraham Z Snyder, Justin L Vincent, Maurizio Corbetta, David C Van Essen, and Marcus E Raichle. The human brain is intrinsically organized into dynamic, anticorrelated functional networks. *Proceedings of the National Academy of Sciences*, 102(27):9673–9678, 2005.

[30] Ward Van Heddeghem, Sofie Lambert, Bart Lannoo, Didier Colle, Mario Pickavet, and Piet Demeester. Trends in worldwide ict electricity consumption from 2007 to 2012. *Computer Communications*, 50:64–76, 2014.

[31] Anders SG Andrae and Tomas Edler. On global electricity usage of communication technology: trends to 2030. *Challenges*, 6(1):117–157, 2015.

[32] Martijn Koot and Fons Wijnhoven. Usage impact on data center electricity needs: A system dynamic forecasting model. *Applied Energy*, 291:116798, 2021.

[33] Anders SG Andrae. Prediction studies of electricity use of global computing in 2030. *International Journal of Science and Engineering Investigations*, 8(86):27–33, 2019.

[34] Truong Vinh Truong Duy, Yukinori Sato, and Yasushi Inoguchi. Performance evaluation of a green scheduling algorithm for energy savings in cloud computing. In *2010 IEEE international symposium on parallel & distributed processing, workshops and Phd forum (IPDPSW)*, pages 1–8. IEEE, 2010.

[35] Xin Li, Zhuzhong Qian, Sanglu Lu, and Jie Wu. Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center. *Mathematical and Computer Modelling*, 58(5-6):1222–1235, 2013.

[36] David Meisner and Thomas F Wenisch. Does low-power design imply energy efficiency for data centers? In *IEEE/ACM International Symposium on Low Power Electronics and Design*, pages 109–114. IEEE, 2011.

[37] Sparsh Mittal and Jeffrey S Vetter. A survey of methods for analyzing and improving gpu energy efficiency. *ACM Computing Surveys (CSUR)*, 47(2):1–23, 2014.

[38] Wuxiong Zhang, Weidong Fang, Qianqian Zhao, Xiaohong Ji, and Guoqing Jia. Energy efficiency in internet of things: An overview. *Computers, Materials & Continua*, 63(2), 2020.

[39] Aimee Van Wynsberghe. Sustainable ai: Ai for sustainability and the sustainability of ai. *AI and Ethics*, 1(3):213–218, 2021.

[40] Iakovina Kindylidi and Tiago Sérgio Cabral. Sustainability of ai: The case of provision of information to consumers. *Sustainability*, 13(21):12064, 2021.

[41] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

[42] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13693–13696, 2020.

[43] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on comFputer vision and pattern recognition*, pages 779–788, 2016.

[44] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

[45] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[46] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021.

[47] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[48] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[49] Mark Buckler, Philip Bedoukian, Suren Jayasuriya, and Adrian Sampson. Eva^2: Exploiting temporal redundancy in live computer vision. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pages 533–546. IEEE, 2018.

[50] Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from jpeg. *Advances in Neural Information Processing Systems*, 31, 2018.

[51] Gaith Rjoub, Jamal Bentahar, Omar Abdel Wahab, and Ahmed Saleh Bataineh. Deep and reinforcement learning for automated task scheduling in large-scale cloud computing systems. *Concurrency and Computation: Practice and Experience*, 33(23):e5919, 2021.

[52] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.

[53] Nina Siu-Ngan Lam, Hong-lie Qiu, Dale A Quattrochi, and Charles W Emerson. An evaluation of fractal methods for characterizing image complexity. *Cartography and Geographic Information Science*, 29(1):25–35, 2002.