CLIPE 2024 - Creating Lively Interactive Populated Environments (2024)
N. Pelechano and J. Pettré (Editors)

LexiCrowd: A Learning Paradigm towards Text to Behaviour Parameters for Crowds

M. Lemonari[1], N. Andreou[1], N. Pelechano[2], P. Charalambous[3], Y. Chrysanthou[1,3]

[1]University of Cyprus, Cyprus
[2]Universitat Politècnica de Catalunya, Spain
[3]CYENS Centre Of Excellence, Cyprus

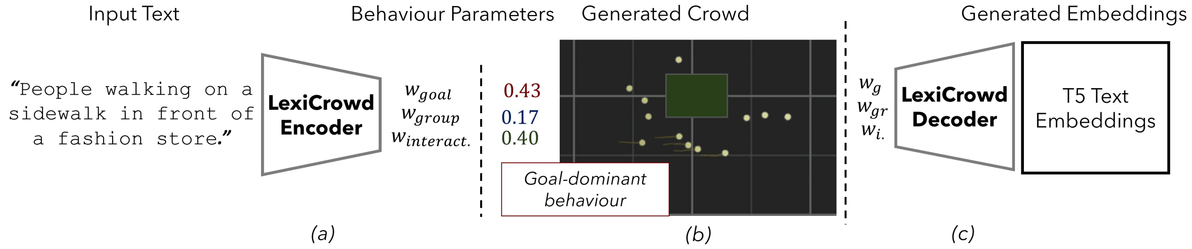


Figure 1: *LexiCrowd Inference Pipeline: Expressive text is fed in the encoder (a) which predicts latent values corresponding to continuous weights of three basic crowd behaviours: goal, group, and interaction with areas of interest. The structured latent space is compatible with simulator parameters (b) allowing text-to-crowd generation. The decoder maps behaviour weights back to textual embeddings (c).*

Abstract

Creating believable virtual crowds, controllable by high-level prompts, is essential to creators for trading-off authoring freedom and simulation quality. The flexibility and familiarity of natural language in particular, motivates the use of text to guide the generation process. Capturing the essence of textually described crowd movements in the form of meaningful and usable parameters, is challenging due to the lack of paired ground truth data, and inherent ambiguity between the two modalities. In this work, we leverage a pre-trained Large Language Model (LLM) to create pseudo-pairs of text and behaviour labels. We train a variational auto-encoder (VAE) on the synthetic dataset, constraining the latent space into interpretable behaviour parameters by incorporating a latent label loss. To showcase our model's capabilities, we deploy a survey where humans provide textual descriptions of real crowd datasets. We demonstrate that our model is able to parameterise unseen sentences and produce novel behaviours, capturing the essence of the given sentence; our behaviour space is compatible with simulator parameters, enabling the generation of plausible crowds (text-to-crowds). Also, we conduct feasibility experiments exhibiting the potential of the output text embeddings in the premise of full sentence generation from a behaviour profile.

CCS Concepts

• ***Computing methodologies*** → ***Neural networks;*** *Natural language processing; Computer graphics;*

1. Introduction

Understanding and generating realistic crowd behaviours is crucial across various domains, including urban planning, virtual environments and the entertainment industry. By understanding and replicating the semantics of how crowds move and interact, creators can craft dynamic scenes that add authenticity and depth to storytelling, resonating with audiences and elevating the quality of entertainment media. However, manually tuning crowd parameters to achieve precise crowd behaviours can be a cumbersome process. Thus, creators opt for intuitive control such as natural language descriptions, which enables them to define diverse and realistic crowd characteristics, behaviours, and interactions, efficiently and at a larger scale. Using text to guide the simulation is intuitive, making it usable by novice and expert users, and alleviating the burden of defining specific and detailed control signals; a sophisticated behaviour could be challenging to define but easy to describe in writing. Hence, aligning the textual and crowd parameter space

DOI: 10.2312/cl.20241049

unlocks new creative opportunities and facilitates alternative behaviour analysis processes.

The inherent ambiguity between language and crowd behaviours makes direct text-to-crowd dynamics generation a particularly challenging task. Besides, the lack of paired data restrains the possibility of training machine learning models in a supervised manner. Previous methods either use simplified annotations, or process the sentences in a rule-based manner to extract behaviour-related attributes ([RPP21, All10, KFS*16]). Recent advancements in the text-to-image and related domains, bear the possibility of using an intermediate modality e.g., images for a text-image-crowd pipeline. Even so, images do not clarify the ambiguity between text and movements; most systems describe images or videos of crowds with generic wording e.g., people walking. Hence, a more representative intermediate modality is behaviour weights, able to reflect subtle text changes on the resulting crowds.

Here we simplify the problem of text-to-crowd generation and approach it with a holistic view. The current progressive simulation methods enabled diverse agent profiles generation from a set of "crowd parameters". We propose an additional component which serves as a mapping from natural language to the crowd parameters. The complete framework gives us the ability to generate crowds from text via behaviour profiles.

To achieve our goal, we design a novel framework which leverages the power of pre-trained language models. We integrate an encoder/decoder module that maps textual embeddings to a constrained, meaningful parameter space for collective behaviours. The structured latent space can be used to simulate crowds, based on an input, short description by generating the compatible simulator parameters. By sampling the latent space we obtain embedded text representations, revealing dormant potential for training specific crowd-related Natural Language Processing (NLP) tasks.

Our contributions can then be summarised as follows:

- We formulate a novel paradigm for the task of text-to-crowd generation which utilises an intermediate representation, namely behaviour parameters inferred from text. We utilise a LLM to create pseudo-pairs of text descriptions and crowd parameters.
- We propose a mechanism, capable of mapping expressive textual prompts to parameters compatible with a crowd simulator (crowd parameters).
- We demonstrate the ability to generate textually controlled plausible crowds, and the potential to infer novel sentences from our output embeddings.

2. Related Work

Our model makes use of pre-trained language models for text-to-crowd generation. We review the literature on language models and related works which inspired the text-to-crowd parameters task. We also discuss past work that use semantics or short sentences to guide the generation process.

2.1. Language models

Large language models (LLMs) are rapidly shaping multiple computer vision tasks due to findings which support their ability to effectively represent semantics.

Pre-trained models such as BERT [DCLT18] and RoBERTa [LOG*19], rely on transformers and are trained in an unsupervised manner using random word masking. These models can be further fine-tuned for particular tasks. Radford et al. [RNS*18] introduce Generative Pre-Trained Transformer (GPT), which adopts task-aware transformations during fine-tuning and succeeds in numerous subsequent language tasks with minimal modifications to the original pre-trained model. Raffel et al. [RSR*20] demonstrate a unified design for all text-based language tasks, T5, which converts all tasks to text-to-text format. We opt for this design as our pre-trained embedding model in order to enable our framework to be extensible to subsequent text-to-crowd parameter tasks. Finally, towards the direction of open and efficient foundation language models, Touvron et al. [TLI*23] introduce LLaMA, trained exclusively on open-source data, permitting its use across a wider range of domains.

Extending beyond language models, several works investigate the connection of language with other modalities. Radford [RKH*21] et al. introduce CLIP, a model which paved-the way for semantic understanding between vision and language. CLIP is trained using contrastive learning with the objective of drawing together the latent spaces of text and images which correspond semantically, while it pushes away those which do not.

Language models have also been used in 3D generation. In particular, Michel et al. [MBOL*22] introduce Text2Mesh, a framework used to stylize 3D meshes using language guidance, while Hong et al. [HZP*22] introduce AvatarCLIP for text-driven generation of 3D avatars. Several works [CLK*23,TGH*22] make use of CLIP embeddings to guide 3D motion generation. Inspired by the above multimodal applications, we introduce LexiCrowd, which aims to bridge the gap between natural language and parameters which can be used to generate collective crowd behaviours.

2.2. Text-based control for crowds

Research on crowd simulation leveraging recent technological advancements, proposed various ways to intuitively author collective behaviours with direct control such as simple behaviour sliders [PKL*22] and sketch-based interface [CvTH*20], or more implicit controls like trajectory data [JCP*10] or images [LPP*]; for a more detailed discussion on authoring tools, refer to [LBC*22]. Still, the huge potential of language models and outstanding results on similar tasks, as well as the pursuit of quick, flexible and straightforward ways for untrained creators to influence their crowds, motivated the domain of text-based control.

Early works mostly include simple text ques such as labels, semantic annotations, or short structured sentences. Seminal work in this is the CAROSA system [All10] that enables high-level control of agents' schedules and responsibilities via an intuitive Miscosoft Office tool. Kapadia et al. [KFS*16] supports the creation and processing of storyboards generating complete narratives, by employing parameterised behaviour trees and partial planners. ACUMEN's influence maps [KBK16] propose an environment-centric way of driving the simulation and create grouping and heterogeneity. Following the semantics-to-crowds angle, Rogla et al. [RPP21] generates crowds procedurally, by generating agent agendas based on rule-based grammars and a semantically-augmented environment. These methods achieve generations of diverse-looking populations, but lack the creative freedom and direct control as most of them are based on specified rules and a limited use of natural language.

Closer to the text-to-crowds concept, is the family of methods handling sentences as the input. Most works ([CWL20, MNC*21, LWC20]) are able to process full sentences, influencing however, the crowd animations, which is outside of the scope of behaviour generation. For example Liu et al. [LWC20] extract essential crowd elements from data structures representing the input sentences. It is worth mentioning that industry-related softwares (Maya Golaem Tool, Unity [Uni, Aut, Goa]) integrate randomness in the distribution process of their crowd parameters to diversify the results e.g., spawn/goal, speed etc.

The limited work on crowd authoring via natural language descriptions demonstrates both the urgency and difficulty of achieving a high-quality, representative result. Our work aims to take a step further in this direction, showcasing its potential and carve future directions.

3. Methodology

Our learning paradigm consists of two main components: the creation of a paired synthetic dataset, and the design of the LexiCrowd architecture.

3.1. Synthetic paired dataset

To train the VAE, we build a dataset of roughly 30,000 labelled sentences, which we generate using OpenAI's `gpt3-5-turbo-instruct` [Ope]. `Gpt3-5-turbo-instruct` is a text-to-text generation model that allows the user to input a textual instruction according to which the text generation process is guided; the model is built to interpret and execute instructions. We craft three types of instructions (Table 1) to guide a categorised sentence generation. Based on the generated sentence's inputted instruction, we assign one of three labels that signify collective behaviours that are dominant in either goal-seeking, grouping, or interaction with areas of interest. The choice of labels was inspired by the selection in Panayiotou et al. [PKL*22]; the aim is to constrain the latent space to an interpretable parameter space like the one developed in [PKL*22]. Hence, we gather a set of sentences, each labelled either "goal", "group", or "interaction". By design, we generate ten sentences with each instructed model run, as a way to enforce textual variety among our data, which is essential for the generalisability of the VAE later on. Note that, the choice of text generation model and exact wording of instructions, was made on a thorough trial-and-error basis until generation was reasonable enough according to our judgement and expectations. We provide some examples of generated sentences for each label type:

- **Goal Ex.1**:"Students were rushing towards their classrooms, hoping to make it to their lectures on time."
- **Goal Ex. 2**:"The commuters briskly walked towards their respective bus stops."
- **Group Ex. 1**: "A group of tourists huddled together, following closely behind their guide as they explored the crowded market."
- **Group Ex.2**: "Families scattered across the park, some playing frisbee, others having a picnic, but all enjoying the warm summer day."
- **Interaction Ex. 1**: "A group of tourists gathered around the historic monument, snapping photos and discussing its significance."
- **Interaction Ex. 2**:"A crowd formed in front of the street performer, mesmerised by his impressive juggling skills."

Label	Instruction
Goal	"In the context of people moving in public, generate 10 sentences revealing goal-oriented behaviours for most people."
Group	"In the context of people moving in public, generate 10 sentences revealing grouping behaviours and movements for most people."
Interaction	"In the context of observing movements in public, generate 10 sentences revealing people mostly interacting with areas of interest."

Table 1: *The label-instruction correspondence used to generate labelled synthetic sentences.*

3.2. Model architecture

An overview of the LexiCrowd architecture is shown in Figure 2. The model's latent space is defined by a continuous space of behaviour parameters corresponding to weighted values of each dominant behaviour i.e., $\{w_{goal}, w_{group}, w_{inter.}\}$. The model encodes text to the behaviour weights' latent space ($\{\mu, \sigma\}$]). The latent space is sampled to obtain a behaviour profile $\mathbf{w}$, which is then passed through the decoder which moves from latent space to the textual embedding space. We train the model with the three losses:

- Label Loss: a cross entropy loss that enforces the dominant behaviour weight and thus structures the parameter latent space into interpretable values.
- Embedding Loss: a cosine similarity loss between the encoded and decoded textual T5 embeddings.
- KL Divergence (KLD) Loss: matching the learned latent distribution to a predefined prior distribution.

Encoder: Our sentences are pre-processed using standard NLP practices before being fed into the encoder. We use the T5 tokenizer, and `max_seq_len = 47` for padding; 47 was the maximum length of all tokenized sentences used for training/testing.

Pre-trained T5 Encoder: We then train our VAE utilising the encoding power of a large, existing language model ("T5ForConditionalGeneration"). We use the pre-trained text encoder of `google/flan-t5-large` [CHL*22] that takes the processed data, applies an embedding layer followed by a transformer encoder, and outputs its last hidden state. We refer to this

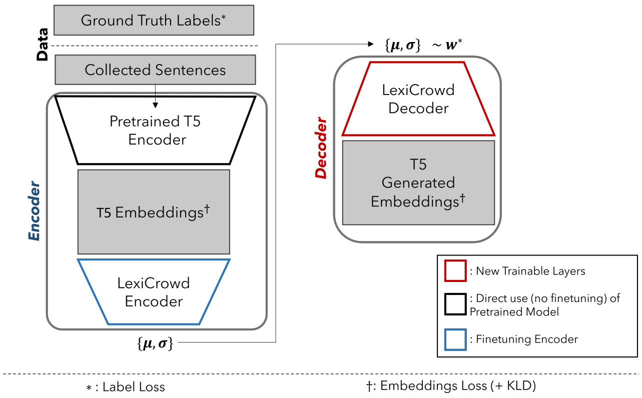


Figure 2: *Model Architecture.*

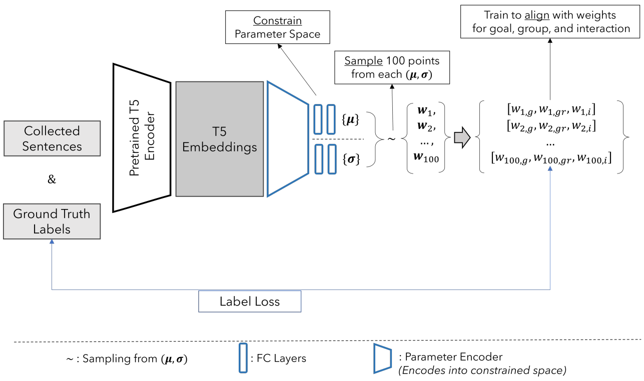


Figure 3: *Preliminary Encoder Architecture.*

as the T5 embedding space since we do not allow additional training of the T5 encoder, and therefore there is a direct, unchangeable link between the input embeddings and the T5 encoder output. T5 is a powerful transformer-based text-to-text generation model, with 783M parameters, trained on three datasets gsm8k, lambada, aqua_rat [CKB*21, PKL*16, LYDB17]) for a number of tasks like translation and question-answering. Using its encoder alleviates the need to train an embedding layer and transformer encoder ourselves, which would require large amounts of data and computational resources; T5 is publicly available and accessible. In short, we use a pre-trained text encoder to go from our input data to the T5 embedding space.

LexiCrowd Encoder: Having obtained the textual embeddings, we feed them into the LexiCrowd encoder to obtain six values corresponding to three-dimensional mean μ and variance σ (`log_var`). First, we train this without the decoder (i.e., without the embedding and KLD loss), and then fine-tune it during the complete encoder/decoder training. Figure 3 shows the architecture and training strategy of this encoder. We note that we pre-train this model as a plausibility test (i.e., to assess whether we can indeed constrain the latent space as intended), and make the encoder/decoder training more time-efficient. During the preliminary training, we sample multiple points from a single distribution, and train on the average loss. The encoder is comprised of three fully-connected layers with dedicated layers to both mean and variance. Before finetuning, the model achieves 89.60% accuracy and 0.3056 loss on test data; the data was split into 80%/20% train and test.

LexiCrowd Decoder: We sample a point from the parameter space (encoder output) and pass it through the decoder that outputs a generated T5 embedding vector. The decoder consists of fully-connected layers with hidden and latent dimensions 1024 and 1024∗ `max_seq_len`, respectively. We then apply the embedding and KLD losses on the decoder outputs.

3.3. Training

All models have been trained using an 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz, 2304 Mhz, 8 Core(s), 16 Logical Processor(s) CPU, 16GBs of RAM and an NVIDIA GeForce RTX 3070 GPU.

To regulate the relative importance between the individual losses and after experimentation, we formulate the following weighted loss function:

$$loss = loss_{KLD} + loss_{emb.} + 200 * (loss_{label}(\mathbf{w}) + loss_{label}(\mu))$$

The model was trained on four epochs with batch size 128, learning rate 0.001, and yielding a 4.39 and 78% embedding reconstruction loss and label accuracy respectively, on the test set.

3.4. Simulator

The trained LexiCrowd encoder outputs three values which are designed to correspond to behaviour weights i.e., goal-seeking, grouping, and interaction with areas of interest, which are compatible with an existing crowd simulator i.e., the modified version of CCP as presented in Lemonari and Panayiotou's P2C [LPP*], originally proposed in [PKL*22]. This choice was intentional so that we could visualise the predicted behaviours easily and efficiently. Also, the encoder latent space spans a wide range of crowd behaviours, allowing mixture of the three distinct aforementioned dominant behaviours which in turn captures more intricate behavioural patterns; the chosen simulator is capable of directly visualizing such behaviours. Here, we emphasise that LexiCrowd could support other simulators via training on different parameter spaces compatible with the desired simulators; LexiCrowd is a learning paradigm and some of its components are interchangeable, requiring however, retraining. Although the inherent simulator limitations impact the final visualizations of LexiCrowd, a study of optimal simulator parameter space and capabilities is outside of the scope of this work.

4. Experiments and Results

The aim of this work is to use LexiCrowd for two applications, illustrated in Figure 1. Hence, our experiments concern the following model functionalities:

- *Firstly*, mapping any input text to a structured parameter space for crowd simulation ($\{w_g, w_{gr}, w_i\}$).

- *Secondly*, enabling generation of novel text embeddings by sampling the learned parameter space according to an input sentence, hence guiding the generation towards a contextually similar embedded sentence.

4.1. Paradigm applicability to real data

For the purpose of examining the transfer-ability of the paradigm to real data, we collect user descriptions of real crowds. We use LexiCrowd to predict the crowd parameters for the descriptions given by real users, and map them to dominant behaviours (labels). We analyse the results in terms of different datasets and modalities.

4.1.1. Real data collection

Given the fact that our model was trained on AI-generated text, we speculate there are still some intangible differences with "real" text. We therefore conduct a user study to collect sentences corresponding to how people describe collective movement in text. We ask users to give textual descriptions with emphasis on collective behaviours, given visual stimuli. Specifically, the users are exposed to four modalities of stimuli, all depicting collective movements: *(1)* a single image, *(2)* a set of three consecutive images, *(3)* a real video, and *(4)* a synthetic video corresponding to the real trajectories. All stimuli come from four datasets of captured crowds corresponding to four different environments, namely a church yard, a University campus, a tram stop, and a commercial road outside a clothing store ([LCL07, CKGC14, PESVG09]). We use abbreviated versions of the names i.e., "Church", "Uni", "ETH", and "Zara". Each participant was randomly assigned to groups that showed each scenario in a different modality; Figure 4 shows snippets of the stimuli.

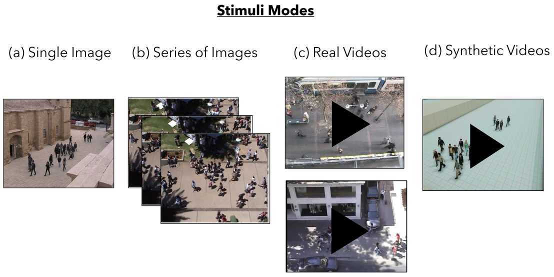


Figure 4: *User study stimuli modalities. For demonstration purposes, here we show examples of different scenarios: Church (a), Uni (b), ETH/Zara (c), and Zara (d).*

During the study, users are asked to describe the behaviour they observe, for each visual stimuli, via the instruction: **"Please, describe what is happening in this sample in text - one or two sentences. Imagine that you want to recreate this based on the description you give"**. Note that, instructing actual people to describe collective behaviours requires more caution since we cannot assume users' knowledge of specific terminology (e.g., trajectories), or intuition (e.g., the meaning of collective behaviours in the context of graphics/crowd simulation). We had a total of 34 participants, diverse in terms of nationality and sex. We collected a total of 235 sentences, 33 being image descriptions, 33 series of images descriptions, 103 and 66 real and synthetic video descriptions, respectively. The objective is to use this user data as inputs to our trained model in order to explore its capabilities in-the-wild, on unseen and real text.

4.1.2. Labelling real data

Firstly, we explore the labelling capabilities of our model i.e., using the encoder to assign dominant behaviours to unseen input sentences. Therefore, we apply our model on the collected user descriptions (described in Section 4.1.1), which are given for four datasets (Zara, Uni, ETH, and Church), across four modalities (single image, series of images, real video, synthetic video); for each dataset we show two or three short videos corresponding to different clips of the full data. For each obtained description, our model predicts the dominant behaviour as well as the corresponding behaviour weights.

Averaging the outputs over all modalities and video clips, our models predicts grouping dominant behaviours for the majority of the data points (descriptions). Table 2 shows the breakdown of the label predictions among the different scenes. This result means that the majority of the user-given sentences for each respective dataset, are predicted to describe a group-dominant behaviour. For the Uni and Church datasets, this is expected since they depict dense student populations. However, it is surprising for Zara and ETH, since they depict pedestrians. To assess this further, we limit the inputs to only real video descriptions. Figure 5 (a), shows the label distributions for real video inputs only. We notice an increased variation across datasets, suggesting that different modalities may yield different described behaviours; we examine this in more detail in Section 4.1.3.

Dataset	Goal Label	Group Label	Inter. Label
Zara	29%	48%	22%
Uni	29%	55%	15%
ETH	29%	47%	24%
Church	27%	54%	19%

Table 2: *Distribution of predicted user data labels.*

Speculating that different clips contain different behaviours and thus yield different descriptions, we further isolate each clip in Figure 5 (b). This reveals a variety between clips e.g., Uni Clip 2 does not include any object interaction, and ETH Clip 2 has a high percentage of goal labels, substantiating some of our expectations. Church Clip 1 being interaction dominant, contrary to Clip 2, could be explained since the group of students show more interest in the church when they first arrive (Clip 1). Note that, the same users described each pair clips of Uni, ETH, and Church, which may have impacted what they paid attention to in the second clip. Zara clips were distributed among the three user groups and so are described by different users. We see that in Zara Clip 2, there is high interaction compared to Clips 1 and 3. Here we remind that the model depends on the text and so does not "see" the videos/images. So, sentences in Zara Clip 2 such as *"People walking in the street by a shop window, with bags. Some show interest in the shop's contents"* and *"People are walking in a shopping district of a city,* *some of them are stopping in front of windows to look at clothes in the shops"* encourage a stronger interaction prediction.

Our model encoder not only predicts the labels, but also the behaviour weights e.g., a set of weights $\{0.5, 0.2, 0.3\}$ represents a goal dominant behaviour but also includes secondary ones. We process the output so that the weights have positive values, summing up to 1. A plot of the Church scenario points is shown in Figure 6. We also plot the standard deviations of the predicted weights in Figure 5 (a2).We can see that Zara and ETH have the most and least diverse predictions, respectively, especially for the interaction weight. The Church high interaction variance can be attributed to the fact that after some time student proceed to the yard uninterested in the building. However, the high Zara variance is not so intuitive. The small number of participants enables one user's response to impact the statistics and given that this study included mostly non-experts, descriptions are highly dependent on user criteria e.g., some users not referring to environment. Obtaining real, ground truth pairs of descriptions and behaviour weights is the gateway to a stronger evaluation.

4.1.3. Modality analysis

As motivated earlier, another interesting analysis direction is to look for differences across the stimuli modalities. For these modalities, it is safe to assume that:

- Single images cannot communicate movement fully, but rather, the intention of it.
- Short series of images reveal short-term movements, but not global goals.
- Synthetic videos (of the type that we used) lack environment and other context information e.g., people's demeanor.

Comparing the predicted weight means of each modality, hinted in Figure 7, suggests:

- Image vs Video: Lower goal weights on average, which is sensible as goal is a long-term movement.
- Set of Images vs Video: No particular pattern found compared to videos. We observe a higher grouping in sets that include stationary groups i.e., Uni compared to moving groups i.e., Church. This is expected since consecutive images reveal static behaviour easier that moving behaviours.
- Synthetic vs Real Video: Lower interaction weights on average, which is reasonable due to the absence of virtual environment.

4.2. Qualitative results of text-to-crowds

One of the significant functionalities of our model is being able to generate simulations from textual descriptions. This is done either via obtaining the behaviour weights from the distribution mean directly (encoder output), or by sampling the learned latent space for novel weights that have embedding similarities with the input text. Our full model was trained end-to-end, and for generating crowd parameters we employ the learned encoder only. We obtain the generated weights and simulate using Lemonari and Panayiotou et al.'s version of CCP [LPP*, PKL*22].

First, we check the plausibility of our model by inputting custom sentences which are designed to describe our three dominant behaviour choices. We pass these controlled sentences into the model and plot the sampled latents along with snapshots of the latent-driven simulation in Figure 8; the generations are feasible.

Next, we input in-the-wild text (no controlled sentences) from the user study responses. Table 3 provides text-parameters examples of the following inputs; the sampled points were averages over 5 model runs.

- ETH Input: "People waiting and walking down a tramway or train station." (Figure 9).
- Zara Input: "People walking in the street by a shop window, with bags. Some show interest in the shop's contents."
- Church Input: "High school students walking in a Greek orthodox church yard."
- Uni Input: "Top down video of many groups of people of 2 to 6 people walking, talking and sitting, in a wide pedestrian street."

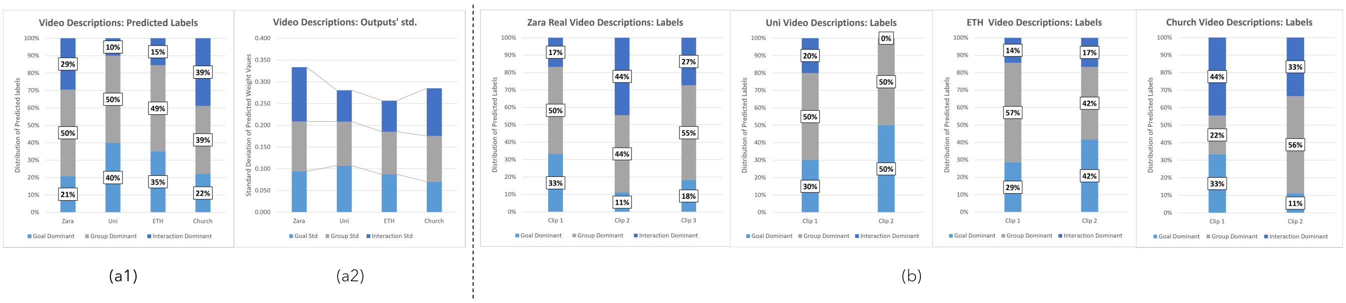
Input	Distribution Mean	Sampled Point
ETH Input	$\{\mathbf{0.49}, 0.35, 0.27\}$	$\{\mathbf{0.47}, 0.31, 0.22\}$
Zara Input	$\{0.20, 0.22, \mathbf{0.58}\}$	$\{0.34, 0.18, \mathbf{0.48}\}$
Church Input	$\{0.20, 0.32, \mathbf{0.48}\}$	$\{0.17, \mathbf{0.48}, 0.35\}$
Uni Input	$\{0.25, \mathbf{0.52}, 0.23\}$	$\{0.19, \mathbf{0.61}, 0.20\}$

Table 3: *Generated behaviour weights on the format:*$\{w_g, w_{gr}, w_i\}$.

We find that our model generates plausible crowd behaviours, and the novel weights produced by the sampling, give rise both to similar behaviours as the inputs, as well as more interesting alternatives. We remind that the scope of our work does not include the simulation process from parameters to crowds (we use an existing tool). Even though the latent space is compatible with the simulator parameters (modified CCP), we observe that the tool requires relatively higher goal weight to motivate the people to move. In our latent space even a group-only description includes movement. Finetuning a simulator to align entirely with the learned latent space will further improve simulation quality.

4.3. Embedding space analysis

The second component of LexiCrowd is the decoder that establishes a mapping between the latent space (behaviour weights) and formatted text embeddings (T5 embedding format). This implies the existence of a embeddings-to-sentences stream. Given this connection, we choose to only output T5 embeddings; generating tokens would require training/using a transformer decoder on a specific NLP task and hence either choosing one of T5 approved tasks e.g., translation, summarisation, or needing ground truth data. Therefore, in order to evaluate our model's potential we conduct a series of experiments with a list of sentences as candidate generations and so bypassing the need for transformer decoder and inverse embedding layers. Specifically, we obtain the T5 encoder embeddings of both the input and the candidates, then apply our model to get a novel, semantically-similar T5 embedding vector, and finally compare the cosine similarities between the novel and candidate embeddings. Table 4 shows examples of this, and the corresponding chosen candidate as "generated sentence" based on the LexiCrowd output. For instance, we check if there is a match between the input



(a1) (a2) (b)

Figure 5: *(a) User data statistics across datasets, averaged for all modalities and all clips. (b) Predicted labels for each video clip.*

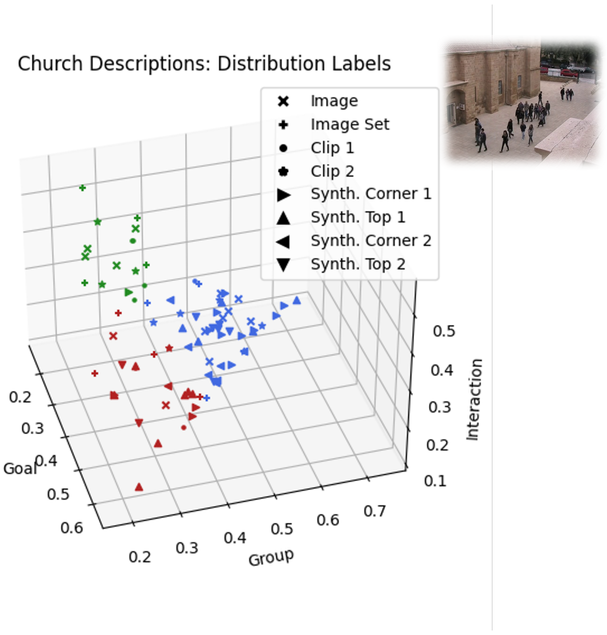


Figure 6: *Visualising predicted weights for each user response of the Church scene.*

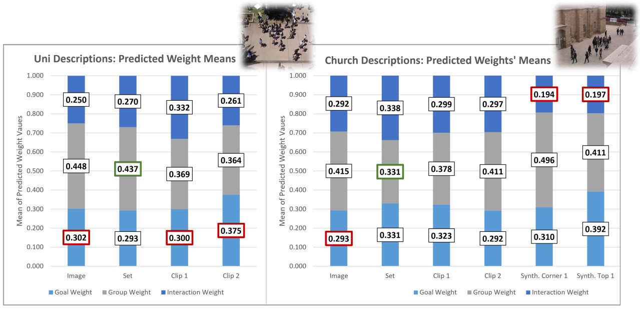


Figure 7: *Modality comparison of predicted behaviours; detected patterns are confirmed with the Zara/ETH results as well.*

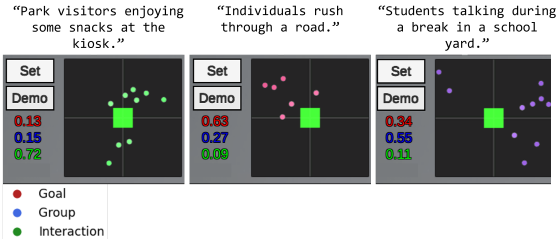

“Park visitors enjoying some snacks at the kiosk.”
“Individuals rush through a road.”
“Students talking during a break in a school yard.”
Set
Demo
0.13
0.15
0.72
0.63
0.27
0.09
0.34
0.55
0.11
Goal
Group
Interaction

Figure 8: *Custom, controlled input sentences along with predicted behaviour weights/simulation snapshots.*

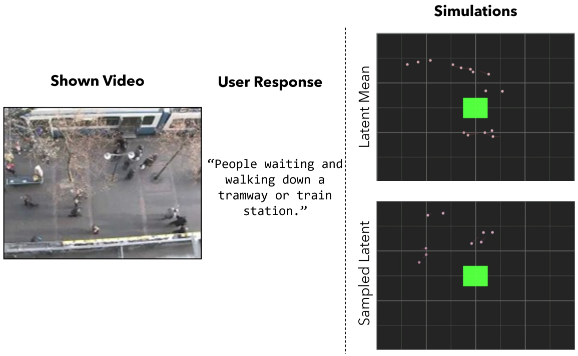

Simulations
Shown Video
User Response
“People waiting and walking down a tramway or train station.”
Latent Mean
Sampled Latent

Figure 9: *Example of encoder results.*

and candidates when candidates talk about the same topic but one of them is in the writing style of the input. The results are promising in regards to using our model to train a transformer decoder for specific NLP tasks e.g., stylised generation, behaviour re-targeting, social interaction understanding, and context recognition.

5. Discussion

In a nutshell, we presented a learning paradigm that can facilitate the alignment of the text and behaviour parameter spaces. LexiCrowd's architecture leverages a LLM and integrates an encoder/decoder scheme that constrains and structures the parameter space into meaningful latents (weights for goal, group, and interaction). Our model encoder was used to classify user-gathered descriptions of four different datasets (zara, uni, ETH, church), into goal-dominant, group-dominant, and interaction-dominant (interaction with areas of interest). The findings reveal a predomi-

Guide text	Candidate-based generations
Stylised Generation:	
People attending a concert.	**1. Students going to class.** 2. The students have been going to class. 3. The students were heading towards the classrooms. 4. The classroom would soon be filled with students.
behaviour Re-targeting:	
People walking in a park.	1. People standing at the park. 2. People eating near the park kiosk. 3. People walking in a mall. **4. People walking in a market.**
Social Interaction Understanding:	
People walking rapidly towards a concert.	1. People are friendly. **2. People are distant.** 3. People are aggressive.
Context Recognition:	
People walk towards a concert.	1. Business professionals walk towards a concert. **2. Music enthusiasts walk towards a concert.** 3. An angry mob walks towards a concert.

Table 4: *Sentence similarity in terms of crowd behaviour attributes. The bold sentence is the closest match to the input, among the given options.*

nant agreement of LexiCrowd's predictions with our anticipations. Surely, training on real sentences would benefit the generalisability on in-the-wild descriptions. Having said that, designing such a collection pipeline is ambiguous. Even the small user study responses are unstructured and subject to users' knowledge of simulating collective behaviours.

The model can also be used for text-to-crowd generation by obtaining the latent parameters that match behaviour weights which can be used as simulator inputs (for a specific simulator). The generations are qualitatively plausible; specifically, for controlled inputs, we get feasible behaviours. However, currently, there are no ground truth behaviour weights to properly evaluate our model. Quantitatively assessing the simulations with trajectory-based comparisons with the real data implies the simultaneous evaluation of the simulator, which is outside the scope of this work. Notably, the current simulator has a weight imbalance i.e., equal weights (0.3) result in mostly grouping behaviours. A possibility for the future is to fine-tune the simulator parameters to be fully aligned with the learned latent space structure. Still, the generation capabilities of our model is an endeavor that parameterises natural language into intuitive and plausible behaviours.

The decoding module delivers novel textual embeddings with semantic similarities to the input text via latent space sampling. Currently, our framework does not handle the process of translating the embedded text to a reconstructed sentence. The choice of our produced encodings structure requires a transformer decoder for full token generation. Nevertheless our model demonstrates potential for full sentence generation; we show that the decoder embeddings hold significant semantic information with regards to the input. Training a transformer decoder on specific NLP tasks e.g., social interaction understanding, would maximise our framework's functionalities and deliver a text-parameters-text pipeline; the challenge lies in obtaining ground truths for such tasks.

Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska Curie grant agreement No 860768 (CLIPE project).

References

[All10] Allbeck J. M.: Carosa: A tool for authoring npcs. In *Motion in Games: Third International Conference, MIG 2010, Utrecht, The Netherlands, November 14-16, 2010. Proceedings 3* (2010), Springer, pp. 182–193. 2

[Aut] Autodesk: AUTODESK Maya. https://www.autodesk.com/products/maya/overview?term=1-YEAR&tab=subscription. Accessed: 2021-10-27. 3

[CHL*22] Chung H. W., Hou L., Longpre S., Zoph B., Tay Y., Fedus W., Li Y., Wang X., Dehghani M., Brahma S., et al.: Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416* (2022). 3

[CKB*21] Cobbe K., Kosaraju V., Bavarian M., Chen M., Jun H., Kaiser L., Plappert M., Tworek J., Hilton J., Nakano R., Hesse C., Schulman J.: Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168* (2021). 4

[CKGC14] Charalambous P., Karamouzas I., Guy S. J., Chrysanthou Y.: A data-driven framework for visual crowd analysis. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 41–50. 5

[CLK*23] Chen R., Liu Y., Kong L., Zhu X., Ma Y., Li Y., Hou Y., Qiao Y., Wang W.: Clip2scene: Towards label-efficient 3d scene understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 7020–7030. 2

[CvTH*20] Colas A., van Toll W., Hoyet L., Pacchierotti C., Christie M., Zibrek K., Olivier A.-H., Pettré J.: Interaction fields: Sketching collective behaviours. In *MIG 2020: Motion, Interaction, and Games* (2020). 2

[CWL20] Chen C.-Y., Wong S.-K., Liu W.-Y.: Generation of small groups with rich behaviors from natural language interface. *Computer Animation and Virtual Worlds 31*, 4-5 (2020), e1960. 3

[DCLT18] Devlin J., Chang M.-W., Lee K., Toutanova K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018). 2

[Goa] Goalem: Golaem. https://golaem.com/. Accessed: 2021-10-27. 3

[HZP*22] Hong F., Zhang M., Pan L., Cai Z., Yang L., Liu Z.: Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *arXiv preprint arXiv:2205.08535* (2022). 2

[JCP*10] Ju E., Choi M. G., Park M., Lee J., Lee K. H., Takahashi S.: Morphable crowds. *ACM Transactions on Graphics (TOG) 29*, 6 (2010), 1–10. 2

[KBK16] Krontiris A., Bekris K. E., Kapadia M.: Acumen: Activity-centric crowd authoring using influence maps. In *Proceedings of the 29th International Conference on computer animation and social agents* (2016), pp. 61–69. 3

[KFS*16] Kapadia M., Frey S., Shoulson A., Sumner R. W., Gross M. H.: Canvas: computer-assisted narrative animation synthesis. In *Symposium on computer animation* (2016), pp. 199–209. 2

[LBC*22] Lemonari M., Blanco R., Charalambous P., Pelechano N., Avraamides M., Pettré J., Chrysanthou Y.: Authoring virtual crowds: A survey. *Computer Graphics Forum 41*, 2 (2022), 677–701. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14506, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14506, doi:https://doi.org/10.1111/cgf.14506. 2

[LCL07] Lerner A., Chrysanthou Y., Lischinski D.: Crowds by example. In *Computer graphics forum* (2007), vol. 26, Wiley Online Library, pp. 655–664. 5

[LOG*19] Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V.: Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019). 2

[LPP*] Lemonari M., Panayiotou A., Pelechano N., Kyriakou T., Chrysanthou Y., Aristidou A., Charalambous P.: P2c: A paths-to-crowds framework to parameterize behaviors. doi:10.36227/techrxiv.170654693.38725484/v1. 2, 4, 6

[LWC20] Liu W.-Y., Wong S.-K., Chen C.-Y.: A natural language interface with casual users for crowd animation. *Computer Animation and Virtual Worlds 31*, 4-5 (2020), e1965. 3

[LYDB17] Ling W., Yogatama D., Dyer C., Blunsom P.: Program induction by rationale generation: Learning to solve and explain algebraic word problems. *ACL* (2017). 4

[MBOL*22] Michel O., Bar-On R., Liu R., Benaim S., Hanocka R.: Text2mesh: Text-driven neural stylization for meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 13492–13502. 2

[MNC*21] Mainardi G., Normoyle A., Cassol V., Badler N., Musse S. R.: An authoring tool to provide group and crowd animation using natural language scripts. In *2021 20th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)* (2021), IEEE, pp. 153–161. 3

[Ope] OpenAI: gpt3.5-turbo-instruct. https://platform.openai.com/docs/models/overview. Accessed: 2024-02-15. 3

[PESVG09] Pellegrini S., Ess A., Schindler K., Van Gool L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision* (2009), IEEE, pp. 261–268. 5

[PKL*16] Paperno D., Kruszewski G., Lazaridou A., Pham N. Q., Bernardi R., Pezzelle S., Baroni M., Boleda G., Fernandez R.: The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Berlin, Germany, August 2016), Association for Computational Linguistics, pp. 1525–1534. URL: http://www.aclweb.org/anthology/P16-1144. 4

[PKL*22] Panayiotou A., Kyriakou T., Lemonari M., Chrysanthou Y., Charalambous P.: Ccp: Configurable crowd profiles. In *ACM SIGGRAPH 2022 Conference Proceedings* (New York, NY, USA, 2022), SIGGRAPH '22, Association for Computing Machinery. URL: https://doi.org/10.1145/3528233.3530712, doi:10.1145/3528233.3530712. 2, 3, 4, 6

[RKH*21] Radford A., Kim J. W., Hallacy C., Ramesh A., Goh G., Agarwal S., Sastry G., Askell A., Mishkin P., Clark J., et al.: Learning transferable visual models from natural language supervision. In *International conference on machine learning* (2021), PMLR, pp. 8748–8763. 2

[RNS*18] Radford A., Narasimhan K., Salimans T., Sutskever I., et al.: Improving language understanding by generative pre-training. 2

[RPP21] Rogla O., Patow G. A., Pelechano N.: Procedural crowd generation for semantically augmented virtual cities. *Computers & Graphics 99* (2021), 83–99. 2, 3

[RSR*20] Raffel C., Shazeer N., Roberts A., Lee K., Narang S., Matena M., Zhou Y., Li W., Liu P. J.: Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research 21*, 1 (2020), 5485–5551. 2

[TGH*22] Tevet G., Gordon B., Hertz A., Bermano A. H., Cohen-Or D.: Motionclip: Exposing human motion generation to clip space. In *European Conference on Computer Vision* (2022), Springer, pp. 358–374. 2

[TLI*23] Touvron H., Lavril T., Izacard G., Martinet X., Lachaux M.-A., Lacroix T., Rozière B., Goyal N., Hambro E., Azhar F., et al.: Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023). 2

[Uni] Unity: Unity Documentation Blend Trees. https://docs.unity3d.com/Manual/class-BlendTree.html. Accessed: 2021-10-27. 3