ORIGINAL ARTICLE

Mitigating exposure bias in large language model distillation: an imitation learning approach

Andrea Pozzi[1] · Alessandro Incremona[1] · Daniele Tessera[1] · Daniele Toti[1,2]

Received: 3 August 2024 / Accepted: 27 February 2025

Abstract
Knowledge distillation is recognized as a valuable model compression strategy that alleviates the computational burden of large language models while preserving performance. This strategy involves training a smaller model utilizing both real data and predictions from a more cumbersome model. Traditional distillation methods, however, are often compromised by exposure bias, which results from reliance on next-step prediction training loss. This bias emerges when models are tested in free-running mode, differing from their training regime and leading to a progressive drift in input distributions between testing and training phases. An analogous issue, known as 'distributional shift', has been effectively addressed in imitation learning through various methodologies. Therefore, this paper specifically tailors an imitation learning-based solution to a traditional knowledge distillation framework which inherently considers both real data and the teacher's predictions as dual sources of expert demonstrations. The effectiveness of this approach is demonstrated over five different test datasets, where it outperforms traditional benchmarks across all evaluation metrics. Specifically, it achieves superior results in perplexity, multi-token generation, and G-Eval score, indicating improvements in both predictive accuracy and alignment with human judgment in text quality. These results underscore the potential of this approach to effectively address exposure bias in large language model distillation.

Keywords Knowledge distillation · Exposure bias · Imitation learning · Multi-token generation

1 Introduction

Text generation has emerged as a particularly significant area within artificial intelligence, driven by its growing array of applications including text summarization, question answering, and neural machine translation. These developments have spurred the scientific community to innovate more effective techniques capable of producing coherent text that mirrors human quality [1]. Currently, the state of the art is represented by transformer architectures [2], which are sophisticated deep neural networks that utilize attention mechanisms. These models significantly surpass the capabilities of earlier methods based on recurrent neural networks (RNNs), particularly in enhancing context understanding across extensive texts.

Among transformer architectures, large language models (LLMs) are characterized by an extensive number of parameters—ranging from billions to trillions—and are trained on massive datasets. The training of these models typically unfolds in an unsupervised manner, focusing on predicting the next word in sequences extracted from a vast collection of textual corpora. A prominent example is the generative pre-trained transformer (GPT) series [3]. Due to the high computational demands of LLM training, several pre-trained versions of these models are made available in open-source format [4], facilitating broader access and experimentation.

https://doi.org/10.1007/s00521-025-11162-0

Published online: 24 March 2025

The high-computational cost associated with LLMs affects both the training and inference phases, limiting their deployment on platforms without high-performance hardware. This challenge underscores the necessity of exploring efficient model compression techniques that minimally impact performance [5], as simply resorting to a smaller model often results in a performance decline. To this end, knowledge distillation offers a promising solution [6]. Within this framework, a less resource-intensive 'student' model learns not only to predict the next word in a dataset, but also to emulate the behavior of a more complex 'teacher' model. This method aims to retain the teacher model's proficiency while reducing the computational demand. In recent years, several approaches have been proposed in the literature for performing knowledge distillation in LLMs [7].

Training the student model within the traditional distillation framework typically involves a single-step optimization process. This method introduces a phenomenon known as exposure bias [8], which occurs when the model is used to generate text autoregressively over multiple steps. Specifically, during free-running mode generation, the model uses words sampled from its own predictions as inputs. As a result, the distribution of test inputs can diverge significantly from that seen during training, largely due to the accumulation of prediction errors as the generation sequence lengthens. While some researchers have questioned the severity of this issue [9], numerous studies affirm its substantial impact [10], identifying it as a primary cause of repetitiveness and hallucinations in language models during extended text generation [11].

Recently, exposure bias has been associated with distributional shift within the framework of imitation learning [12], a subfield of reinforcement learning where an intelligent agent learns to mimic the optimal behavior of an expert based on provided demonstrations. In imitation learning, distributional shift occurs when an agent trained on expert demonstrations deviates from the expert's behavior during deployment, encountering unfamiliar states. This shift arises because, during training, the agent follows the expert's distribution of states and actions. However, when generating its own actions, it may veer into untrained areas, compounding its errors over time. In this area, several strategies have been developed to mitigate such phenomena [13–15]. Notably, these techniques have also been adapted for time series prediction [16], where ground truth data are treated as expert demonstrations to address exposure bias effectively.

The primary contribution of this manuscript is the introduction of an innovative methodology to address the issue of exposure bias in language modeling within the original framework of knowledge distillation, as introduced by [6]. The approach in [6] uses both the teacher's predictions (soft labels) and ground truth data (hard labels) to train the student model. In contrast, existing imitation learning methods for mitigating exposure bias, such as *ImitKD* [17], typically rely only on soft labels. To address exposure bias in LLM distillation while retaining the benefits of hard labels, this paper adapts and combines two imitation learning techniques—Dataset Aggregation (DAgger) [15] and Data as Demonstrator (DaD) [16]—to effectively leverage both label types as complementary sources of expert demonstrations. To evaluate the robustness of our approach, extensive experiments have been conducted on a dataset taken from the novel *A Song of Ice and Fire*, as well as on textual corpora consisting of science fiction TV show plot summaries, demonstrating consistent improvements across diverse datasets. Results indicate that enhanced performance can be achieved when contributions from both soft and hard labels are considered, thus underscoring the benefit of including the latter in reducing exposure bias. Our method outperforms conventional next-step prediction techniques and other well-known imitation learning-based approaches, such as *ImitKD*, across three metrics: perplexity, which measures next-token prediction accuracy; free-running cross-entropy, which evaluates the model's capability to generate coherent, multi-token sequences in an autoregressive manner; and G-Eval score, a state-of-the-art, prompt-based metric that assesses the alignment of generated text quality with human judgment [18].

The rest of the paper is organized as follows. Section 2 reviews related work on exposure bias in sequence prediction and knowledge distillation. Section 3 formally defines the knowledge distillation problem for LLMs and examines how exposure bias arises in this context. Section 4 introduces imitation learning, covering key theoretical concepts that provide the necessary background for understanding the proposed methodology. Section 5 details the proposed approach to mitigate exposure bias in LLM distillation. Section 6 describes the case study setup, evaluation metrics, and empirical results showcasing the effectiveness of the proposed strategy.

Finally, Sect. 7 concludes the paper, summarizing the findings and suggesting potential directions for future research.

2 Related work

Various strategies have been proposed to address exposure bias, particularly in RNNs and language generation tasks. This section surveys existing approaches for mitigating exposure bias in both sequence prediction and language model distillation, highlighting their contributions and limitations.

2.1 Exposure bias in sequence prediction

A foundational approach to addressing exposure bias in RNNs was introduced by [19] with a method called *scheduled sampling*, which modifies the training algorithm by stochastically replacing true input tokens with those generated by the model. Although empirically valuable, *scheduled sampling* has been criticized due to its improper objective function, which leads to an inconsistent learning algorithm that outputs a biased estimator [20].

To overcome this issue, the authors in [21] presented a training methodology called *professor forcing*, leveraging the principles of generative adversarial networks (GANs) [22]. However, this methodology faces significant challenges, including the added complexity of integrating a discriminator and the known difficulties associated with training GANs, such as training instability that can lead to mode collapse and convergence issues [23].

Another significant contribution was introduced by the authors in [24], which trained a reinforcement learning agent to generate sequences of tokens using its own predictions as inputs and optimizing sequence-level metrics. Similarly, an actor-critic solution was proposed by [25], while an offline reinforcement learning methodology was discussed in [26]. Another reinforcement learning-based approach was presented in [27] to mitigate exposure bias by relying on a stepwise reward function derived directly from a model trained with teacher forcing. A technique from the field of inverse reinforcement learning was used in [28] to train a language model to match token sequences instead of individual token likelihoods, with the aim of alleviating exposure bias. Finally, a methodology based on contrastive learning was proposed by the authors in [29] to mitigate exposure bias in neural machine translation.

However, these works primarily address sequence prediction and do not directly pertain to knowledge distillation, which is the focus of this paper.

2.2 Exposure bias in language model distillation

Although exposure bias has been recognized in the context of language model distillation [30–32], efforts to directly address it have been scarce.

The authors in [17] proposed the so-called *ImitKD* algorithm, which employs techniques from imitation learning to explicitly mitigate exposure bias when training an RNN student model to mimic a teacher. However, *ImitKD* focuses only on matching the predictions provided by the teacher, which may not be sufficient for the student model to achieve high performance, as pointed out in [6]. In fact, traditional knowledge distillation implies the inclusion of an additional objective in the optimization to predict the ground truth data from a specific transfer set, to attain satisfactory text generation. A similar limitation is observed in the approaches proposed by [33] and [34], where the knowledge distillation framework is formulated as an optimization of generalized divergence measures. Likewise, hard labels are not utilized in the solution proposed by [35], which employs a reinforcement learning-based approach to mitigate exposure bias in knowledge distillation by enabling student exploration.

In summary, the proposed methodology builds on these concepts and introduces a mixed imitation learning strategy to address exposure bias in LLM distillation, leveraging ground truth data and the teacher's outputs, which are both deemed essential for achieving high performance, as highlighted in the seminal work by [6].

3 Problem description

This section examines the issue of exposure bias in language model distillation. In particular, Sect. 3.1 introduces the foundational concepts of language models and knowledge distillation, while Sect. 3.2 investigates the causes and consequences of exposure bias.

3.1 Language modeling and knowledge distillation

In language modeling, a model p_θ is tasked with parameterizing the conditional probability $p(w_t|w_{1:t-1})$ of a token $w_t \in \mathcal{V}$ following a sequence of preceding tokens $w_{1:t-1} = \{w_1, \ldots, w_{t-1}\}$, where $\mathcal{V}$ is the vocabulary. Given the complexity of conditioning on the entire history of tokens, a Markov assumption is typically employed:

$$p(w_t|w_{1:t-1}) \approx p(w_t|w_{t-H:t-1}) \tag{1}$$

for all $w_t \in \mathcal{V}$, where H is referred to as the context size.

Training a model p_θ on a text corpus $\mathcal{D}$ involves minimizing a specific loss function with respect to the model parameters θ. Commonly, the negative log-likelihood is adopted:

$$L_{\text{NLL}}^{\mathcal{D}}(\theta) = -\frac{1}{|\mathcal{D}|}\sum_{t=H+1}^{|\mathcal{D}|} \log p_\theta(w_t|w_{t-H:t-1}). \tag{2}$$

It is important to note that minimizing this loss function corresponds to solving the so-called next-token prediction task.

A trained model p_θ can be employed to generate a sequence of text by iteratively sampling the next token according to the probability distribution defined by the model, and then shifting the context window to include the new token. Sampling can be performed stochastically, where each token is drawn according to its conditional probability:

$$\hat{w}_t \sim p_\theta(\cdot|\tilde{w}_{t-H:t-1}), \tag{3}$$

with the input sequence initialized as $\tilde{w}_{1:H} = w_{t:H}$ and iteratively updated as follows:

$$\tilde{w}_{t-H:t-1} \leftarrow \{\tilde{w}_{t-H+1:t-1}, \hat{w}_t\}. \tag{4}$$

Alternative sampling strategies include greedy sampling, where the most probable token is consistently selected, and beam search decoding, which expands multiple probable sequences concurrently to find an optimal sequence [36].

Knowledge distillation involves training a smaller model, designated as the student model p_θ, using targets derived not only from a corpus $\mathcal{D}$, referred to as the transfer set, but also from the predictions of a larger model $p^\star$, known as the teacher. As demonstrated by the authors in [6], incorporating the dense vector of the teacher's outputs as a target can significantly enhance the student's performance on the transfer set. The inclusion of the teacher's predictions within the training process results in a loss function that integrates two distinct components

as a convex combination, parameterized by $\alpha \in [0, 1]$. The first component is the negative log-likelihood specified in (2), while the second one represents a distillation term, denoted as $L_{\mathrm{KD}}^{\mathcal{D}}(\theta)$, that penalizes the discrepancy between the student and teacher outputs:

$$L^{\mathcal{D}}(\theta) = \alpha L_{\mathrm{NLL}}^{\mathcal{D}}(\theta) + (1 - \alpha) L_{\mathrm{KD}}^{\mathcal{D}}(\theta). \tag{5}$$

There are various formulations for the distillation loss $L_{\mathrm{KD}}^{\mathcal{D}}(\theta)$, as found in the literature. For example, an L_2 loss between the student and teacher logits is suggested in [37], while the authors in [6] advocate for employing a cross-entropy loss between the soft output distributions of the student and teacher, a method that becomes equivalent to the approach of [37] under high-temperature settings. Other potential loss functions include the Kullback–Leibler divergence or mean squared error between student and teacher output probabilities, though these are generally less effective than regression on logits [37]. Given these considerations, this paper adopts the approach proposed in [6]:

$$L_{\mathrm{KD}}^{\mathcal{D}}(\theta) = -\frac{1}{|\mathcal{D}|} \sum_{t=H+1}^{|\mathcal{D}|} l_t(\theta) \tag{6}$$

with

$$l_t(\theta) = p^{\star}(w_t | w_{t-H:t-1}) \log p_\theta(w_t | w_{t-H:t-1}). \tag{7}$$

The ultimate objective of knowledge distillation is to produce a smaller student model whose performance on a specific domain dataset, referred to as the transfer set, ideally approaches that of the larger teacher model, despite having significantly fewer parameters, thus mitigating the computational costs associated with deploying large-scale models while maintaining high levels of performance.

3.2 Exposure bias

Exposure bias emerges from a fundamental mismatch in the input distributions between the training and deployment phases of text generation models. During training, these models rely on a loss function, as defined in (2), which penalizes errors in next-token prediction based on the true previous sequence of tokens. However, during inference—such as in tasks like text summarization, story generation, or translation—the model generates a sequence by predicting each consecutive token based on the previously generated ones. This discrepancy between training and inference methodologies can lead to degraded performance, as the model's training conditions do not mirror its actual use [19, 21]. In practice, replacing true tokens ($w_{t-H:t-1}$) with predicted ones ($\tilde{w}_{t-H:t-1}$) during inference introduces cumulative errors. As the sequence generation progresses, these prediction errors compound, significantly altering the expected distribution as the generation window extends, exacerbating the mismatch between training and actual deployment scenarios.

4 Imitation learning

The exposure bias discussed in previous sections is intrinsically connected to the concept of distributional shift, a known challenge in the field of imitation learning [38]. This paradigm, frequently employed in system control and regarded as a specialized domain of reinforcement learning, focuses on the process of learning from demonstrations. This approach involves training a learning agent to replicate the behavior of an expert agent within a decision-making framework represented as a Markov decision process (MDP). In such a context, an expert agent

provides demonstrations of optimal actions across various states of the environment and a learning agent observes these demonstrations to learn how to mimic the expert's behavior, ultimately striving to achieve comparable performance. This section aims to shed light on key imitation learning algorithms. Specifically, Sect. 4.1 introduces behavioral cloning, the foundational approach in imitation learning, along with its primary limitation: distributional shift. This subsection also emphasizes the analogy between distributional shift and the previously discussed exposure bias. Sections 4.2 and 4.3 discuss established solutions to this challenge, specifically the DAgger and DaD algorithms, respectively.

4.1 Behavioral cloning and distributional shift

In the imitation learning paradigm, the behavior of an agent is represented by a policy π which, for each time step t, maps the state of the environment s_t into the agent's action a_t. Demonstrations of the optimal policy $\pi^\star$, provided by an expert agent, are assumed to be available to the learner, in the form of K state-action trajectories $\xi_k = \{(s_1, \pi^\star(s_1)), \ldots, (s_{T_k}, \pi^\star(s_{T_k}))\}$, where T_k represents the length of the k-th trajectory, with $k \in \{1, \ldots, K\}$. Therefore, the primary goal for the learner is to develop a policy $\hat{\pi}$ that effectively replicates the expert's decision-making process. The most straightforward approach to train such a learner involves using supervised techniques to directly map states to the expert's actions, optimizing the policy $\hat{\pi}$ in order to minimize the discrepancy between $\pi^\star(s_t)$ and $\hat{\pi}(s_t)$ over the states encountered in the provided demonstrations. This specific approach in the context of imitation learning is termed behavioral cloning. Despite being conceptually simple and widely adopted, behavioral cloning suffers from certain limitations that pave the way for more complex strategies. Among the main limitations of behavioral cloning, the issue of distributional shift emerges. The concept of distributional shift refers to the divergence that arises between the learner behavior during training and its performance in testing environments. During training, the learner closely follows the expert's decisions, ensuring that its state-action distribution remains within familiar bounds. However, during testing or deployment, it must generate its own sequence of actions autonomously. When minor deviations from the expert occur, they can compound over time, leading to unfamiliar states and producing increasingly inaccurate behaviors. A visual representation of this phenomenon is depicted in Fig. 1. This challenge is analogous to the exposure bias observed in language model distillation: In both cases, errors accumulate when the model generates sequences without the benefit of corrective guidance at each step, leading to a drift from the desired output. Given this analogy, more advanced techniques from imitation learning, particularly those aimed at mitigating distributional shift, present a promising avenue for addressing exposure bias in LLM distillation.

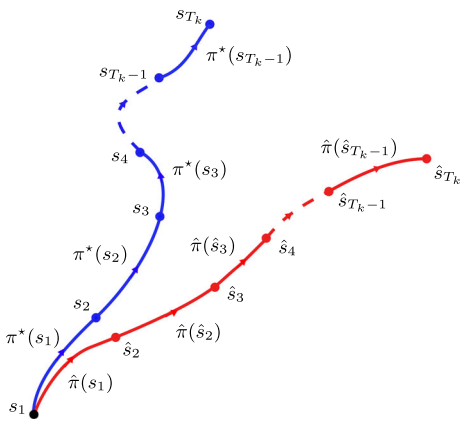


Fig. 1 Visual description of distributional shift: the blue curve represents the state trajectory when the expert policy $\pi^\star$ is applied, while the red one describes the states encountered by the learner policy $\hat{\pi}$, both starting from a common state s_1

4.2 Dataset aggregation (DAgger)

Among the most established solutions proposed to address the problem of distributional shift is the so-called DAgger algorithm, proposed by the authors in [15], which involves regularly updating the training dataset with new samples generated by the learner's current policy.

Within this framework, at each iteration $i = 1, \ldots, n$, the training phase consists of optimizing the policy $\hat{\pi}_i$ on the dataset of available expert demonstrations. Following the training phase, the subsequent data aggregation step involves enlarging the training dataset by incorporating new state-action pairs $(\hat{s}_t, \pi^\star(\hat{s}_t))$. These pairs are collected by deploying the currently trained policy $\hat{\pi}_i$, with $\hat{s}_t$ representing the states visited under this policy, and $\pi^\star(\hat{s}_t)$ denoting the actions that the expert would take in those states.

Essentially, the initial policy $\hat{\pi}_1$ effectively serves as a behavioral cloning model, trained solely on the original set of expert demonstrations. As the iterations proceed, DAgger systematically updates the training dataset by adding states encountered by executing the learner's policy, paired with the expert's corrective actions. This iterative refinement helps align the distribution of states visited by the learner with those experienced by the expert, significantly mitigating the issue of distributional shift.

4.3 Data as demonstrator (DaD)

Given the similarities between distributional shift and exposure bias, it is plausible to employ techniques like DAgger within the context of sequence prediction and generation. One of the primary challenges in this application is the necessity of reconceptualizing the problem in terms of imitation learning, particularly when the identity of the expert is ambiguous. An innovative approach to this issue is presented by the authors in [16], who propose viewing the dataset itself as the source of expert demonstrations. This idea leads to the development of the DaD algorithm, which is grounded in robust theoretical principles and has shown efficacy in mitigating exposure bias in time series forecasting.

In this context, trajectories are conceptualized as sequences of states $\xi_k = \{s_1, \ldots, s_{T_k}\}$, and the purpose is to train a model f to minimize the mean squared error distance between the predicted state $\hat{s}_{t+1} = f(s_t)$ and the actual one s_{t+1}, with the training dataset consisting of the pairs (s_t, s_{t+1}). This formulation assumes that the underlying process is Markovian.

DaD operates through a cyclic two-step process, similarly to DAgger, but with a crucial difference in the data aggregation phase. For each iteration i, and for each trajectory $\xi_k = \{s_1, s_2, \ldots, s_{T_k}\}$, the current model f_i is used to generate a new trajectory $\hat{\xi}_k = \{s_1, \hat{s}_2, \ldots, \hat{s}_{T_k}\}$ autoregressively, starting from the initial state s_1. The training dataset is then augmented with the pairs $(\hat{s}_t, s_{t+1})$ for $t = 2, \ldots, T_k - 1$, considering the next true state as a pseudo-expert demonstration. As the algorithm progresses, the multistep predictions of the model increasingly

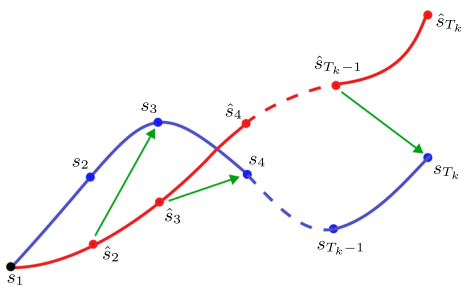


Fig. 2 Schematic representation of the DaD algorithm: the blue line indicates the true state trajectory, while the red line shows the predicted trajectory, both starting from the same initial state s_1. The green arrows can be considered corrections provided by the algorithm to reduce the exposure bias of the model

align with the dynamics observed in the real data, effectively addressing the exposure bias issue. Figure 2 displays a visual representation of the corrective action provided by the DaD algorithm.

5 DaD-DAgger for LLM distillation

Drawing inspiration from the ideas discussed in Sect. 4, the methodology proposed in this work addresses exposure bias in LLM distillation by considering both the data from the transfer set and the teacher's outputs as distinct sources of expert information.

It is worth recalling the distillation framework for training a student model, where true labels from the transfer set are leveraged alongside the dense signal provided by the teacher's outputs. This process involves minimizing the loss function as detailed in (5), which consists of a convex combination of two cross-entropy losses: One on the true labels (see (2)) and the other on the teacher's predicted probabilities (see (6)), with the trade-off regulated by the hyperparameter α. In this framework, the training samples constitute the set of triplets $\mathcal{D}_T = \{(w_{t-H:t-1}, w_t, p^\star(\cdot|w_{t-H:t-1}))\}_{\mathcal{D}}$, used to optimize the student model parameters θ over the corpus $\mathcal{D}$, with $t = H+1, \ldots, |\mathcal{D}|$. The objective is to simultaneously maximize the probability $p_\theta(w_t|w_{t-H:t-1})$ of generating the true token w_t while minimizing the cross-entropy between the learner's predicted probabilities $p_\theta(\cdot|w_{t-H:t-1})$ and the teacher's ones $p^\star(\cdot|w_{t-H:t-1})$.

It is recognized that this setting, which focuses on solving the traditional knowledge distillation problem using a next-token prediction formulation, is susceptible to exposure bias in text generation. To address this, the proposed methodology employs an iterative two-step algorithm comprising a training phase followed by a data aggregation phase (see Algorithm 1), similar to the previously described approaches. Since the training phase of the i-th iteration primarily involves optimizing the parameters θ_i of the student model p_{θ_i} on the available dataset, the subsequent discussion will therefore concentrate on the data aggregation phase.

Algorithm 1 DaD-DAgger Algorithm

Input: Transfer set $\mathcal{D}$, teacher model $p^\star$
Hyperparameters: n and α
$\mathcal{D}_T = \{(w_{t-H:t-1}, w_t, p^\star(\cdot|w_{t-H:t-1}))\}_{\mathcal{D}}$
for $i = 1$ to n **do**
  Train p_{θ_i} on $\mathcal{D}_T$ to minimize (5)
  Initialize $\mathcal{D}_{\text{new}} = \emptyset$
  for each trajectory ξ_k **do**
    Generate trajectory $\hat{\xi}_k$
    for $t = H+1$ to T_k **do**
      Get $(\tilde{w}_{t-H:t-1}, w_t, p^\star(\cdot|\tilde{w}_{t-H:t-1}))$
      Add triplet to $\mathcal{D}_{\text{new}}$
    end for
  end for
  Update $\mathcal{D}_T = \mathcal{D}_T \cup \mathcal{D}_{\text{new}}$
end for
Return refined model parameters θ_n

Consider the transfer set $\mathcal{D}$ segmented into trajectories of T_k tokens:

$$\xi_k = \{w_1, \ldots, w_{T_k}\}. \tag{8}$$

For each iteration i, and for each trajectory ξ_k, a new trajectory $\hat{\xi}_k$ is defined as follows:

$$\hat{\xi}_k = \{w_1, \ldots, w_H, \hat{w}_{H+1}, \ldots, \hat{w}_{T_k}\}, \tag{9}$$

where the initial H tokens match those of ξ_k, and the subsequent tokens are generated by the student model in free-running mode, incorporating its own predictions into the input. Specifically, for $t > H$ the tokens $\hat{w}_t$ are sampled according to the model $p_{\theta_i}(\cdot|\tilde{w}_{t-H:t-1})$. The training dataset is subsequently augmented with the set of triplets $\mathcal{D}_{\text{new}} = \{(\tilde{w}_{t-H:t-1}, w_t, p^{\star}(\cdot|\tilde{w}_{t-H:t-1}))\}_k$ for $t = H+2, \ldots, T_k$.

By continuously updating the training dataset, the student model is provided with additional demonstrations consisting of trajectory corrections, derived from both the true data and the teacher predictions, thus avoiding drift and mitigating exposure bias.

It is important to note that the hyperparameter α regulates the balance between reliance on ground truth data and teacher predictions. When α is closer to 1, the optimization simplifies to a specific case of training on labeled samples, making the method an adaptation of the DaD approach to the LLM context, where only data from the transfer set is used. Conversely, when α approaches 0, the methodology simplifies to a variant of the knowledge distillation framework where only the demonstrations of the teacher are considered, as in *ImitKD*, with the teacher's predicted probabilities directly guiding the learning process.

6 Experimental results

In this section, the main results obtained by applying the proposed methodology are presented. Specifically, in Subsection 6.1, the primary settings of the conducted experiments are illustrated, while Subsection 6.2 is dedicated to the description of the metrics adopted for evaluating the model performance. Finally, Subsection 6.3 validates the methodology at varying degrees of distillation (modulating the hyperparameter α) against both a traditional benchmark that employs next-token prediction, analogous to a behavioral cloning approach in imitation learning, and the *ImitKD* algorithm, whose performance corresponds to those of the proposed approach when $\alpha = 0$.

6.1 Case study description

The student and teacher models utilized in this study are pre-trained causal language models, specifically GPT-Neo 125 M and GPT-Neo 1.3B, respectively, taken from the open-source Hugging Face library [4].

Experiments have been conducted in five separate runs, each utilizing a distinct transfer set. The first transfer set is an extract from the prolog of George R. R. Martin's novel *A Song of Ice and Fire*, containing approximately 135,000 tokens, as processed using the teacher's tokeniser. The remaining four transfer sets consist of plot summaries from episodes of popular science fiction television series, specifically *The X Files*, *Stargate*, *Fringe*, and *Doctor Who*, which have already been used to test the performance of language models by the authors in [39]. After tokenization, these datasets contained 115,193 tokens (*The X Files*), 121,251 tokens (*Stargate*), 80,380 tokens (*Fringe*), and 117,180 tokens (*Doctor Who*). For consistency across all experiments, each dataset has been divided into 80% training, 10% validation, and 10% test partitions.

Since the student model considered in this paper is pre-trained, only a fine-tuning process over the specific transfer set has been conducted, with the training targeting only a subset of the layers, specifically coinciding with the last transformer block. This includes both the multi-head attention and subsequent feed-forward neural network layers, along with the final layer normalization module, which is crucial for stabilizing the output of the last transformer block. It is important to note that the corresponding layers of the teacher model, from which knowledge is distilled, have also been fine-tuned on the same transfer set. The resulting total number of trained parameters in the student model is approximately 7 million, compared to about 50 million in the teacher model.

Although the context size for the pre-trained models under consideration is up to 2048 tokens, the length of the input sequences used in the experiments has been limited to $H = 128$ due to memory constraints on the available infrastructure.

Training has been conducted over 4 epochs, with early stopping applied based on the validation loss. An Adam optimizer has been employed with a learning rate of $2.5 \cdot 10^{-5}$ and a batch size of 32. For the DaD-DAgger algorithm, 3 iterations have been performed, with the number of free-running generated tokens set to 10, resulting in an overall trajectory length of $T_k = 138$.

The hyperparameters were determined during the algorithm's validation phase. The search was conducted within a range of potential values that aligned with the available computational resources, ensuring both the feasibility of the experiments and the reliability of the results. It is worth mentioning that all the experiments have been performed on a HPE ProLiant DL580 Gen10 server equipped with four Intel(R) Xeon(R) Gold 6146 CPU @ 3.20GH, one terabyte of RAM, an NVIDIA Tesla V100 GPU and running Fedora release 35 operating system. The execution of the DaD-DAgger algorithm with the aforementioned setting was required on average about 5.5 GPU hours.

6.2 Metrics

Both the proposed methodology and the next-token prediction algorithm have been evaluated using three metrics to provide a comprehensive assessment of model performance. The first metric is perplexity, computed on the test dataset for both the proposed approach and the benchmark. Perplexity, $P(\theta)$, is defined as:

$$P(\theta) = 2^{\mathrm{CE}(\theta)} \tag{10}$$

where the exponent is the cross-entropy, $\mathrm{CE}(\theta)$, computed on the test dataset using a sequence of H tokens as input to the model and predicting the next token in a one-step approach:

$$\mathrm{CE}(\theta) = -\frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{t=H+1}^{|\mathcal{D}_{\text{test}}|} \log p_\theta(w_t | w_{t-H:t-1}). \tag{11}$$

Perplexity is therefore well-suited to evaluate next-token prediction accuracy in a single-step context but is limited in assessing the model's performance in autoregressive (free-running) text generation [40]. To address this limitation, a second metric, called free-running cross-entropy, has been employed. This metric assesses the model's multistep generation performance by evaluating the cross-entropy between the student model in free-running mode and the true labels over a horizon of $T_{\text{test}} = 10$ generated tokens:

$$\mathrm{CE}_{\mathrm{FR}}(\theta) = -\frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{t=H+1}^{|\mathcal{D}_{\text{test}}|} \log p_\theta(w_t | \tilde{w}_{t-H:t-1}), \tag{12}$$

where $\tilde{w}_{t-H:t-1}$ represents the sequence of tokens generated by the model in free-running mode. Here, the model iteratively samples the next token according to its probability distribution, updating the context window to include each newly generated token. This free-running cross-entropy metric, inspired by methods for evaluating time series predictors in multistep prediction [16], is suited to measure the model's performance in longer text generation scenarios and captures the impact of compounding errors that emerge during open-ended generation. Notably, trends in free-running cross-entropy are not necessarily correlated with those in perplexity, as they evaluate distinct aspects of model performance. To further enhance the evaluation, a third metric has been considered: G-Eval, a prompt-based evaluation designed to capture semantic coherence and stylistic quality in text generation tasks where an exact target text is unavailable. G-Eval employs a LLM (in this paper GPT-4o has

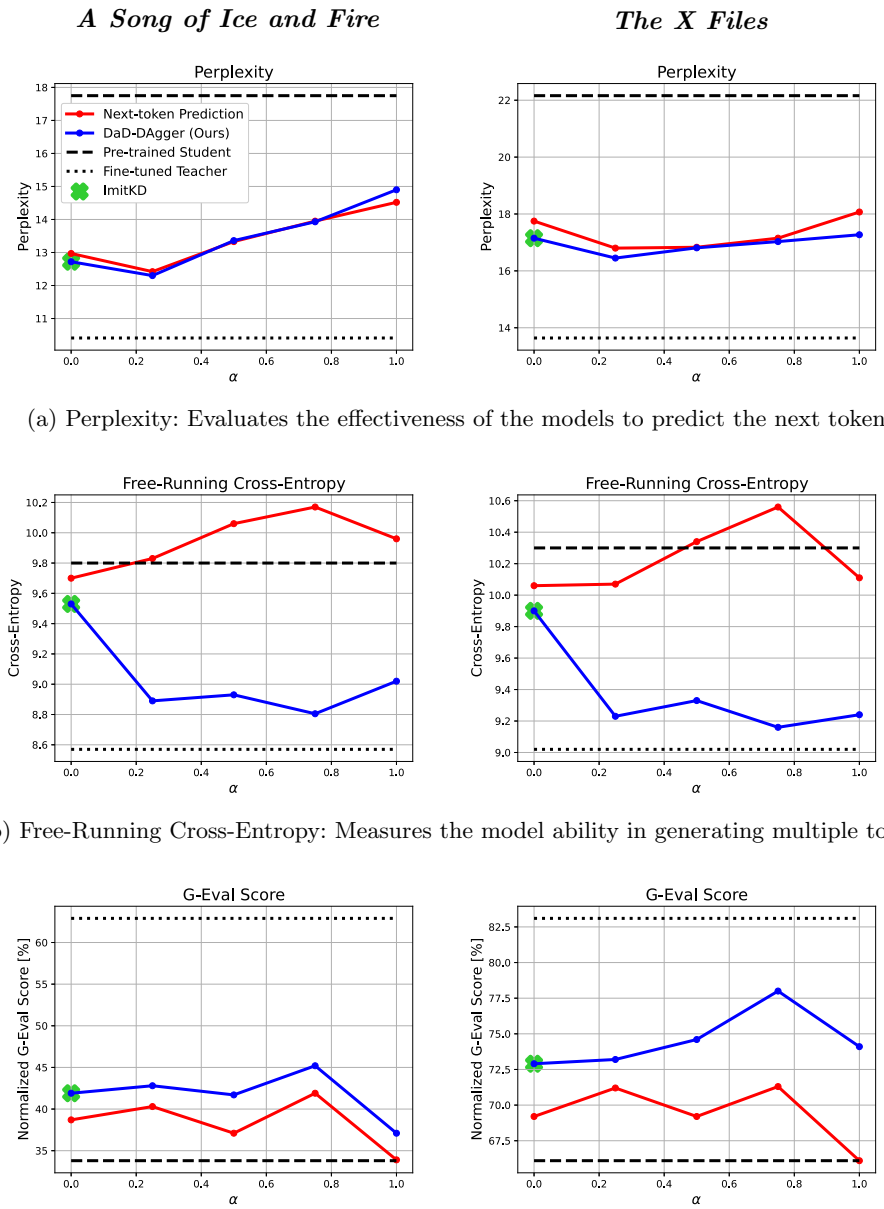
been adopted) and a chain-of-thought scoring approach to assess generated text based on its alignment with the narrative style and coherence of the source material. Specifically, given an input sequence of H tokens and the corresponding continuation of T_{test} generated tokens, G-Eval assigns to the generated text a score between 0 and 1 that reflects both grammatical accuracy and stylistic alignment with the dataset's context [18, 41]. This metric has been chosen for its ability to capture nuanced aspects of storytelling and narrative flow, which are critical for evaluating models in creative text generation tasks [42]. Table 1 provides an overview of the specific criteria used for each dataset. Together, these three metrics provide a holistic evaluation of the student model's performance, assessing next-token accuracy, multistep generative coherence, and alignment with the narrative style.

6.3 Results

In the following, the student model trained using the proposed methodology is compared, based on the previously discussed metrics, for values of $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$, against two benchmarks: the next-token prediction model and the *ImitKD* algorithm (equivalent to the DaD-DAgger algorithm for $\alpha = 0$). Additionally, the performance of the pre-trained student and fine-tuned teacher models are presented for reference purposes.

Table 1 G-Eval criteria used for each dataset in the assessment phase

Dataset	Evaluation Criteria
A Song of Ice and Fire	• Evaluate if the language is similar to the descriptive style of George R.R. Martin.
	• Check whether the themes in the continuation are relevant to a dark, intricate fantasy narrative.
	• Penalize the use of modern language or terminology that does not fit the time period or tone of the story.
	• Check for logical and temporal consistency with the initial prompt, maintaining the tone and style of the provided prompt
The X Files	• Evaluate if the language used reflects the suspenseful, mysterious, and often foreboding tone typical of The X Files.
	• Check whether the themes in the continuation align with The X Files, particularly involving paranormal activity, government conspiracies, extraterrestrial encounters, and unexplained phenomena.
	• Penalize the use of modern slang or concepts that disrupt the show's established setting and its paranormal investigative context.
	• Check for logical and temporal consistency with the initial prompt, maintaining The X Files' tone of procedural investigation mixed with supernatural mystery
Stargate	• Evaluate if the language is similar to the scientific, adventurous tone of the Stargate series.
	• Check whether the themes in the continuation are relevant to a science fiction narrative with interplanetary travel, alien cultures, and technological challenges.
	• Penalize the use of language, technology, or concepts that contradict the established lore and scientific context of the Stargate universe.
	• Check for logical and temporal consistency with the initial prompt, maintaining the tone and style of the Stargate series
Fringe	• Evaluate if the language reflects the investigative, suspenseful, and dark tone typical of Fringe.
	• Check whether the themes in the continuation are relevant to Fringe, involving elements of science fiction, unexplained phenomena, fringe science, and conspiracies.
	• Penalize the use of language or references that break from the show's established scientific and technological context.
	• Check for logical and temporal consistency with the initial prompt, maintaining Fringe's tone of procedural investigation and scientific mystery
Doctor Who	• Evaluate if the language used reflects the adventurous, imaginative, and often eccentric tone typical of Doctor Who.
	• Check whether the themes in the continuation align with Doctor Who, including time travel, alien encounters, moral dilemmas, and cosmic mysteries.
	• Penalize the use of language, technology, or concepts that conflict with the Doctor Who universe or feel anachronistic within its science-fantasy framework.
	• Check for logical and temporal consistency with the initial prompt, maintaining Doctor Who's tone of whimsical adventure combined with underlying gravitas



(a) Perplexity: Evaluates the effectiveness of the models to predict the next token.

(b) Free-Running Cross-Entropy: Measures the model ability in generating multiple tokens.

(c) G-Eval Score: Assesses the quality of generated sequences based on Table 1 criteria.

Fig. 3 Performance comparison of the proposed DaD-DAgger methodology (blue line) on the datasets *A Song of Ice and Fire* (left column) and *The X Files* (right column) against the next-token prediction approach (red line) and the *ImitKD* algorithm defined only for $\alpha = 0$ (green cross). The performance of the pre-trained student (dashed black line) and fine-tuned teacher (dotted black line) are also illustrated

This comparison is shown in Fig. 3, where the plots on the left refer to the *A Song of Ice and Fire* dataset, while those on the right pertain to the dataset from *The X Files*. The plots in Fig. 3a demonstrate that, for each value of α, the perplexity is similar for the proposed methodology and the next-token prediction approach, significantly reduced compared to that of the pre-trained student model. It is evident that the use of soft labels provided by the teacher helps reduce perplexity compared to training with only hard labels ($\alpha = 1$). However, it is also noteworthy that the presence of hard labels is equally necessary; indeed, as α approaches 0 (*ImitKD*), perplexity increases with respect to the minimum achieved in $\alpha = 0.25$. Regarding the free-running cross-entropy, whose variation with respect to α is depicted in Fig. 3b, the proposed methodology achieves significantly lower values for every degree of distillation, compared to the model trained with next-token prediction strategy, which obtains values comparable or slightly higher than the one achieved by the pre-trained student model. Moreover, as it can be noticed from the figure, the DaD-DAgger performance in terms of free-running cross-entropy improve significantly as the parameter α is increased, showing a reduction of about 7% for both the

datasets with respect to the case in which the *ImitKD* algorithm is employed. Finally, Fig. 3c represents the performance of the models in terms of average G-Eval score over the test datasets for different values of α. In order to simplify the interpretation of the results, the average G-Eval score achieved by each model has been normalized with respect to the one obtained using the true continuation of the input. Also in this case, as it can be seen from the figure, the DaD-DAgger algorithm achieves higher performance than the next-token prediction methodology for every value of α. Moreover, it generally improves upon the results obtained by *ImitKD* for values of α greater than 0, with a peak at $\alpha = 0.75$; however, its performance slightly drops below that of *ImitKD* at $\alpha = 1$.

The results presented in Fig. 3 are summarized in Tables 2, 3, and 4, along with the performance metrics across all datasets. These tables provide a comprehensive overview of the next-token prediction and DaD-DAgger algorithms' performance for different values of α, reported in terms of perplexity, free-running cross-entropy, and G-Eval scores. For reference, Table 5 includes metrics for the pre-trained student and fine-tuned teacher models across all datasets, contextualizing both the starting condition and the highest achievable performance. The results indicate that the DaD-DAgger algorithm consistently achieves the best performance across all metrics, with optimal values observed for α between 0.25 and 0.75, closely approximating the outcomes of the fine-tuned teacher model and outperforming the next-token prediction method. These results also highlight the effectiveness of combining hard and soft labels during training, a strategy that outperforms using only soft labels, as implemented in *ImitKD*.

7 Conclusions

In this paper, a new methodology based on imitation learning has been introduced to address exposure bias in the knowledge distillation of LLMs. Considerable effort has been devoted to adapt the imitation learning framework to the general challenges of distillation, wherein the student model is trained using both the output logits from the teacher and the ground truth samples from the transfer set. This adaptation has been done by treating the teacher's predictions and the true data as dual sources of expert demonstrations.

Extensive experiments on five different datasets have been conducted to compare this strategy against traditional supervised and imitation learning approaches, considered as benchmarks. Three metrics have been employed for evaluation: Perplexity, which assesses next-token prediction capabilities; free-running cross-entropy, which evaluates the model's ability to generate coherent, multi-token sequences autoregressively; and G-Eval score, a state-of-the-art, prompt-based evaluation metric that measures alignment with human judgment in generated text quality. The results indicate that the proposed methodology consistently outperforms the benchmarks across all metrics, achieving the highest scores when a blend of hard and soft labels is used in training.

Table 2 Comparison of perplexity achieved by next-token prediction and DaD-DAgger (our approach) across different α values and datasets. Note that the DaD-DAgger algorithm for $\alpha = 0$ corresponds to the so-called *ImitKD* approach

Dataset	Algorithm	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
A Song of Ice and Fire	Next-token prediction	12.97	12.42	13.33	13.95	14.52
	DaD-DAgger	12.72	12.3	13.36	13.93	14.90
The X Files	Next-token prediction	17.75	16.80	16.83	17.15	18.07
	DaD-DAgger	17.15	16.45	16.81	17.03	17.27
Stargate	Next-token prediction	16.68	15.67	15.56	15.66	16.11
	DaD-DAgger	16.02	14.83	15.03	15.45	16.11
Fringe	Next-token prediction	17.39	16.22	16.07	16.01	16.34
	DaD-DAgger	16.91	15.78	15.75	16.06	16.32
Doctor Who	Next-token prediction	17.75	16.68	16.56	16.80	17.51
	DaD-DAgger	17.14	16.11	16.22	16.68	17.39

Table 3 Comparison of free-running cross-entropy achieved by next-token prediction and DaD-DAgger (our approach) across different α values and datasets. Note that the DaD-DAgger algorithm for $\alpha = 0$ corresponds to the so-called *ImitKD* approach

Dataset	Algorithm	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
A Song of Ice and Fire	Next-token prediction	9.71	9.83	10.06	10.17	9.96
	DaD-DAgger	9.53	8.89	8.93	8.81	9.02
The X Files	Next-token prediction	10.06	10.07	10.34	10.56	10.11
	DaD-DAgger	9.90	9.23	9.33	9.16	9.24
Stargate	Next-token prediction	9.53	9.6	9.75	9.93	10.16
	DaD-DAgger	9.41	8.93	9.02	9.04	9.29
Fringe	Next-token prediction	9.74	9.76	9.83	9.95	9.99
	DaD-DAgger	9.66	9.19	9.27	9.06	9.08
Doctor Who	Next-token prediction	9.62	9.66	9.69	9.81	9.93
	DaD-DAgger	9.54	9.00	8.93	9.11	9.14

Table 4 Comparison of normalized G-Eval score achieved by next-token prediction and DaD-DAgger (our approach) across different α values and datasets. Note that the DaD-DAgger algorithm for $\alpha = 0$ corresponds to the so-called *ImitKD* approach

Dataset	Algorithm	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
A Song of Ice and Fire	Next-token prediction	38.7%	40.3%	37.1%	41.9%	33.9%
	DaD-DAgger	41.9%	42.8%	41.7%	45.2%	37.1%
The X Files	Next-token prediction	69.2%	71.2%	69.2%	71.3%	66.1%
	DaD-DAgger	72.9%	73.2%	74.6%	78.0%	74.1%
Stargate	Next-token prediction	41.9%	41.2%	39.2%	38.0%	40.3%
	DaD-DAgger	42.8%	41.5%	39.1%	37.8%	44.1%
Fringe	Next-token prediction	47.5%	44.6%	48.9%	54.7%	56.2%
	DaD-DAgger	52.7%	51.9%	48.6%	57.3%	56.9%
Doctor Who	Next-token prediction	51.8%	53.4%	52.3%	51.8%	49.3%
	DaD-DAgger	53.8%	52.3%	54.8%	51.6%	54.1%

Table 5 Comparison of pre-trained student and fine-tuned teacher performance across different datasets

Dataset	Model	Perplexity	Free-running Cross-entropy	G-Eval
A Song of Ice and Fire	Pre-trained Student	17.75	9.79	33.8%
	Fine-tuned Teacher	10.41	8.57	62.9%
The X Files	Pre-trained Student	22.16	10.3	66.1%
	Fine-tuned Teacher	13.64	9.02	83.1%
Stargate	Pre-trained Student	22.94	10.02	34.3%
	Fine-tuned Teacher	12.47	8.72	59.3%
Fringe	Pre-trained Student	22.01	10.05	43.1%
	Fine-tuned Teacher	13.36	8.84	66.4%
Doctor Who	Pre-trained Student	22.78	9.94	53.6%
	Fine-tuned Teacher	13.74	8.76	63.5%

These findings highlight the method's effectiveness in improving both predictive accuracy and generative quality, underscoring its potential to address exposure bias in LLM distillation techniques.

Future work could explore the application of more sophisticated imitation learning strategies, such as generative adversarial imitation learning (GAIL) [43] and soft-Q imitation learning (SQIL) [44], to further enhance this methodology. Additionally, the ideas proposed by recent studies on data augmentation in knowledge distillation offer promising directions. Specifically, the entropy-based filtering approach suggested in [45] for image classification could be adapted to natural language tasks and potentially integrated into the DaD-DAgger framework as a means to further mitigate exposure bias by improving the quality of expert demonstrations.

Although implementing such a pipeline is beyond the scope of this work, these extensions represent compelling avenues for future research in this area.

Funding Open access funding provided by Università Cattolica del Sacro Cuore within the CRUI-CARE Agreement.

Data Availability Custom code written in Python is available on request. The dataset consisting of an extract of the novel *A Song of Ice and Fire* was downloaded from https://www.kaggle.com/code/shadabhussain/text-generation/input, while the plot summaries from the science fiction TV shows were downloaded from https://paperswithcode.com/dataset/scifi-tv-plots.

Declarations

Conflict of interest The authors have no relevant financial or nonfinancial interests to disclose.

Ethics Approval and Consent to Participate Not applicable.

Consent for publication Not applicable.

References

1. Li J, Tang T, Zhao WX, Nie J-Y, Wen J-R (2024) Pre-trained language models for text generation: A survey. ACM Computing Surveys 56(9):1–39
2. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Advances in neural information processing systems **30**
3. Radford A, Narasimhan K, Salimans T, Sutskever I et al (2018) Improving language understanding by generative pre-training
4. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M et al (2019) Huggingface's transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771
5. Buciluǎ C, Caruana R, Niculescu-Mizil A (2006) Model compression. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 535–541
6. Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531
7. Yang C, Zhu Y, Lu W, Wang Y, Chen Q, Gao C, Yan B, Chen Y (2024) Survey on knowledge distillation for large language models: methods, evaluation, and application. ACM Transactions on Intelligent Systems and Technology
8. Schmidt F (2019) Generalization in generation: A closer look at exposure bias. In: Proceedings of the 3rd Workshop on Neural Generation and Translation, pp. 157–167
9. He T, Zhang J, Zhou Z, Glass J (2019) Quantifying exposure bias for neural language generation
10. Xu Y, Zhang K, Dong H, Sun Y, Zhao W, Tu Z (2019) Rethinking exposure bias in language modeling. arXiv preprint arXiv:1910.11235
11. Wang C, Sennrich R (2020) On exposure bias, hallucination and domain shift in neural machine translation. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3544–3552
12. Arora K, El Asri L, Bahuleyan H, Cheung JCK (2022) Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In: Findings of the Association for Computational Linguistics: ACL 2022, pp. 700–710
13. Daumé H, Langford J, Marcu D (2009) Search-based structured prediction. Machine learning 75:297–325
14. Ross S, Bagnell D (2010) Efficient reductions for imitation learning. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 661–668. JMLR Workshop and Conference Proceedings

15. Ross S, Gordon G, Bagnell D (2011) A reduction of imitation learning and structured prediction to no-regret online learning. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, pp. 627–635. JMLR Workshop and Conference Proceedings
16. Venkatraman A, Hebert M, Bagnell J (2015) Improving multi-step prediction of learned time series models. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29
17. Lin A, Wohlwend J, Chen H, Lei T (2020) Autoregressive knowledge distillation through imitation learning. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 6121–6133
18. Liu Y, Iter D, Xu Y, Wang S, Xu R, Zhu C (2023) G-eval: Nlg evaluation using gpt-4 with better human alignment. arXiv preprint arXiv:2303.16634
19. Bengio S, Vinyals O, Jaitly N, Shazeer N (2015) Scheduled sampling for sequence prediction with recurrent neural networks. Advances in neural information processing systems **28**
20. Huszár F (2015) How (not) to train your generative model: Scheduled sampling, likelihood, adversary? arXiv preprint arXiv:1511.05101
21. Lamb AM, ALIAS PARTH GOYAL AG, Zhang Y, Zhang S, Courville AC, Bengio Y (2016) Professor forcing: A new algorithm for training recurrent networks. Advances in neural information processing systems **29**
22. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. Advances in neural information processing systems **27**
23. Saxena D, Cao J (2021) Generative adversarial networks (gans) challenges, solutions, and future directions. ACM Computing Surveys (CSUR) 54(3):1–42
24. Ranzato M, Chopra S, Auli M, Zaremba W (2016) Sequence level training with recurrent neural networks. In: 4th International Conference on Learning Representations, ICLR 2016
25. Bahdanau D, Brakel P, Xu K, Goyal A, Lowe R, Pineau J, Courville A, Bengio Y (2016) An actor-critic algorithm for sequence prediction. arXiv preprint arXiv:1607.07086
26. Pang RY, He H (2020) Text generation by learning from demonstrations. arXiv preprint arXiv:2009.07839
27. Hao Y, Liu Y, Mou L (2022) Teacher forcing recovers reward functions for text generation. Adv Neural Information Processing Systems 35:12594–12607
28. Wulfmeier M, Bloesch M, Vieillard N, Ahuja A, Bornschein J, Huang S, Sokolov A, Barnes M, Desjardins G, Bewley A et al (2024) Imitating language via scalable inverse reinforcement learning. arXiv preprint arXiv:2409.01369
29. He J, Sun S, Jia X, Li W (2024) Recovery should never deviate from ground truth: Mitigating exposure bias in neural machine translation. In: Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1), pp. 68–79
30. Kim Y, Rush AM (2016) Sequence-level knowledge distillation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1317–1327 (2016)
31. Gu Y, Dong L, Wei F, Huang M (2023) Minillm: Knowledge distillation of large language models. In: The Twelfth International Conference on Learning Representations
32. Calderon N, Mukherjee S, Reichart R, Kantor A (2023) A systematic study of knowledge distillation for natural language generation with pseudo-target training. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 14632–14659
33. Wen Y, Li Z, Du W, Mou L (2023) f-divergence minimization for sequence-level knowledge distillation. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 10817–10834
34. Agarwal R, Vieillard N, Zhou Y, Stanczyk P, Garea SR, Geist M, Bachem O (2024) On-policy distillation of language models: Learning from self-generated mistakes. In: The Twelfth International Conference on Learning Representations
35. Li D, Hao Y, Mou L (2024) Llmr: Knowledge distillation with a large language model-induced reward. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pp. 10657–10664
36. Wiher G, Meister C, Cotterell R (2022) On decoding strategies for neural text generators. Trans the Association for Computational Linguistics 10:997–1012
37. Ba J, Caruana R (2014) Do deep nets really need to be deep? Advances in neural information processing systems **27**
38. Chiang T-R, Chen Y-N (2021) Relating neural text degeneration to exposure bias. In: Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pp. 228–239
39. Ammanabrolu P, Tien E, Cheung W, Luo Z, Ma W, Martin LJ, Riedl MO (2020) Story realization: Expanding plot events into sentences. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 7375–7382
40. Fedus W, Goodfellow I, Dai AM (2018) Maskgan: Better text generation via filling in the _. In: International Conference on Learning Representations
41. Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, Le QV, Zhou D (2022) Chain-of-thought prompting elicits reasoning in large language models. Adv neural information processing systems 35:24824–24837
42. Fu J, Ng S-K, Jiang Z, Liu P (2023) Gptscore: Evaluate as you desire. arXiv preprint arXiv:2302.04166
43. Ho J, Ermon S (2016) Generative adversarial imitation learning. Advances in neural information processing systems **29**

44. Reddy S, Dragan AD, Levine S (2019) Sqil: Imitation learning via reinforcement learning with sparse rewards. arXiv preprint arXiv:1905.11108
45. Wang H, Lohit S, Jones MN, Fu Y (2022) What makes a“ good” data augmentation in knowledge distillation-a statistical perspective. Adv Neural Information Processing Systems 35:13456–13469

Authors and Affiliations

Andrea Pozzi[1] · Alessandro Incremona[1] · Daniele Tessera[1] · Daniele Toti[1,2]

✉ Andrea Pozzi
andrea.pozzi@unicatt.it

Alessandro Incremona
alessandro.incremona@unicatt.it

Daniele Tessera
daniele.tessera@unicatt.it

Daniele Toti
daniele.toti@unicatt.it

[1] Department of Mathematics and Physics, Catholic University of the Sacred Heart, Via della Garzetta 48, Brescia 25133, Italy

[2] Department of Sciences, Roma Tre University, Viale Guglielmo Marconi 446, Roma 00146, Italy