现代大型语言模型（LLMs）和视觉-语言模型（VLMs）在广泛的应用中取得了显著的成功，这得益于大规模预训练的力量。它们随后被微调用于例如机器翻译、对话系统、医疗诊断和多模态推理等任务。尽管这些模型表现出色，但它们仍容易生成事实不正确或有偏见的输出，这通常是由于存在无关、误标记或不具代表性的训练数据所致。

这突出表明需要可扩展的方法来量化特定训练数据点的影响。值得注意的方法包括影响函数 (?) 和基于 Shapley 值的方法 (?)，它们提供了估计单个数据点如何影响模型预测的框架 (??)。这些方法已被证明在检测错误标记数据 (??)、识别有影响力的例子、诊断偏差 (?) 和审计数据集 (?) 等下游应用中效果显著。然而，由于依赖于 Hessian 和重复再训练，对于大模型来说，影响函数和 Shapley 值方法在计算上是不可行的。

为了缓解影响估计的高计算成本，已经引入了几种近似技术。TracIn (?) 通过跟踪训练检查点之间的梯度相似性来估计数据影响，而 DataInf (?) 和 HyperInf (?) 则专注于高效的赫塞矩阵近似。然而，这些方法都涉及显著的权衡：TracIn 需要存储大量的模型快照；DataInf 存在与模型规模相关的近似误差；HyperInf 假设梯度独立并带来三次方的计算复杂度。与此同时，对于 Shapley 值的近似，? 提出了一个在线方法，该方法在训练过程中测量验证数据与训练数据之间的梯度或赫塞矩阵相似性。然而，由于需要在每个训练步骤计算和存储每个样本的梯度，将这一方法应用于单个验证样本仍是不切实际的。关键的是，所有这些方法都依赖于访问模型的梯度和微调后的权重——这些资源在实际的 LLM 和 VLM 部署中常常是无法获取的。其他策略，例如在生成图像模型中使用的基于相似性的方法 (?)，由于它们的序列生成过程，对 LLM 和 VLM 的适用性较差。在这项工作中，我们介绍了一种专为预训练的 LLMs 和 VLMs 量身定制的前向数据估值框架 For-Value 。For-Value 不依赖于梯度或模型重训练，而是根据训练示例对提高验证数据可能性的贡献来估算其价值。受到预训练模型生成丰富且信息丰富的隐藏表示的启发 (???)，我们提出了一种封闭形式的度量，捕捉训练和验证样本之间的表示相似性和预测误差对齐。这个度量使得 For-Value 仅使用一次前向传播就能够识别出有影响力或标记错误的数据，从而具有高度的可扩展性和实用性。我们的贡献如下：

- 我们提出了 For-Value ，这个仅向前的框架用于在将预训练的 LLMs 和 VLMs 适应到下游任务时识别有影响力或嘈杂的训练数据。
- 我们从理论上证明，在监督学习目标下，可以通过隐藏表示和预测误差的对齐可靠地近似影响得分。
- 我们通过实验证明，For-Value 在检测有影响力和标签错误的样本方面与基线方法相匹配或优于基线方法，并且效率提高了多个数量级。

1 相关工作

预训练的大型语言模型（LLM）和视觉-语言模型（VLM）。在现代的机器学习工作流程中，使用预训练的基础模型（FM）并将其适应于特定的下游任务是标准做法 (??)。基础模型，如大型语言模型和视觉-语言模型，由于其在大规模数据集上的广泛预训练，成为强大的初始化点。LLM，包括 LLaMA (?) 和 GPT-4 (?)，是在多样的文本数据上进行训练，以进行语言理解和生成。VLM，如 Qwen2.5-VL (?)、LLaMA-VL (?) 和 GPT-4V (?)，整合视觉和文本输入以执行像图像标注和视觉问答这样的任务。

数据估值。影响估计是一种广泛采用的技术，用于量化单个训练样本对模型预测的影响。(?) 引入了一种基于 Hessian 的方法，通过利用二阶导数来计算影响函数；然而，这种方法会产生难以承受的计算成本，特别是对于大型模型如 LLMs。为了解决这一限制，DataInf (?) 和 HyperInf (?) 提出了有效的近似方法，避免了计算或逆转 Hessian 矩阵的需要，提供了可扩展的影响估计，减少了开销。然而，所有这些基于影响函数的方法都需要先对模型进行微调。另一方面，TracIn (?) 通过在训练检查点中跟踪一阶梯度来采用无 Hessian 方法估计数据影响，但这需要存储和访问许多检查点，对于大型模型来说不切实际。除了基于影响的方法之外，基于 Shapley 值的技术 (?) 通过边际贡献评估数据的重要性。虽然理论上很吸引人，但由于需要反复训练模型，这些方法计算成本昂贵。为了缓解这一问题，? 在训练期间通过度量验证和训练梯度之间的相似性提出了一种在线 Shapley 值近似。然而，将这种方法扩展到单个数据点仍不切实际，因为这需要在每个训练步骤计算和存储每个样本的梯度。与这些方法相反，我们的方法既不需要微调模型也不需要反向传播。

2 预备知识

我们研究一个预训练的大型语言模型（LLM）或视觉-语言模型（VLM），记为 π_θ ，其中 θ 代表其参数。对于给定的输入 $\boldsymbol{x}$ ——这可能由文本标记、图像块或两者结合组成——模型在输出文本序列 $\boldsymbol{y} = (y_1, y_2, \ldots, y_{|\boldsymbol{y}|})$ 上定义了一个条件概率分布，分解为：

$$\pi_\theta(\boldsymbol{y}|\boldsymbol{x}) = \prod_{k=1}^{|\boldsymbol{y}|} \pi_\theta(y_k|\boldsymbol{x}, \boldsymbol{y}_{<k}),$$

其中 $\boldsymbol{y}_{<k} = (y_1, \ldots, y_{k-1})$ 。在每个步骤中，模型预测下一个标记 y_k ，其条件为输入 $\boldsymbol{x}$ 和前缀 $\boldsymbol{y}_{<k}$ 。这种自回归结构是现代大多数用于文本生成 (?)、图像字幕生成 (?) 和多模态推理 (?) 的 LLM 和 VLM 的基础。在本节中，我们提供了一个理论基础，以理解单个训练样本如何影响预训练 LLM 和 VLM 的行为。这些见解激励了我们提出的方法 For-Value 的设计。符号：设 $\boldsymbol{W}$ 、$\boldsymbol{w}_z$ 和 $\boldsymbol{h}_{\boldsymbol{z}}$ 分别表示标记反嵌入矩阵、标记 $z \in \mathcal{V}$ 的反嵌入，其中 $\mathcal{V}$ 是词汇表，以及生成标记 $\boldsymbol{z} \in \mathcal{V}^*$ 的隐藏嵌入，其嵌入维度为 d 。设 $\boldsymbol{z}_k$ 为 k -th 个标记在 $\boldsymbol{z}$ 中，$\boldsymbol{z}_{<k}$ 为 $\boldsymbol{z}$ 中的前 $k-1$ 个标记。最后，用 $\boldsymbol{e}_z \in \mathbb{R}^{|\mathcal{V}|}$ 表示对应于 $z \in \mathcal{V}$ 的标准基向量。给定一个训练数据集 $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^n \in \mathcal{D}$ 和一个估值样本 $(\boldsymbol{x}_v, \boldsymbol{y}_v) \in \mathcal{D}$ ，我们定义数据值如下。

Definition 1 (Data Value). 在任何训练时间 $t > 0$ ，如果一个训练样本导致给定数据点 $(\boldsymbol{x}_v, \boldsymbol{y}_v)$ 的似然性变化 $\frac{d}{dt} \ln \pi_{\theta(t)}(\boldsymbol{y}_v|\boldsymbol{x}_v)$ 更大，那么它被认为对这个数据点更有价值。

数据价值的定义是基础性的，因为它代表了一个训练模型自信地预测给定数据 $(\boldsymbol{x}_v, \boldsymbol{y}_v)$ 的可能性。更普遍地，在现实世界的应用中，我们希望模型在测试环境（例如

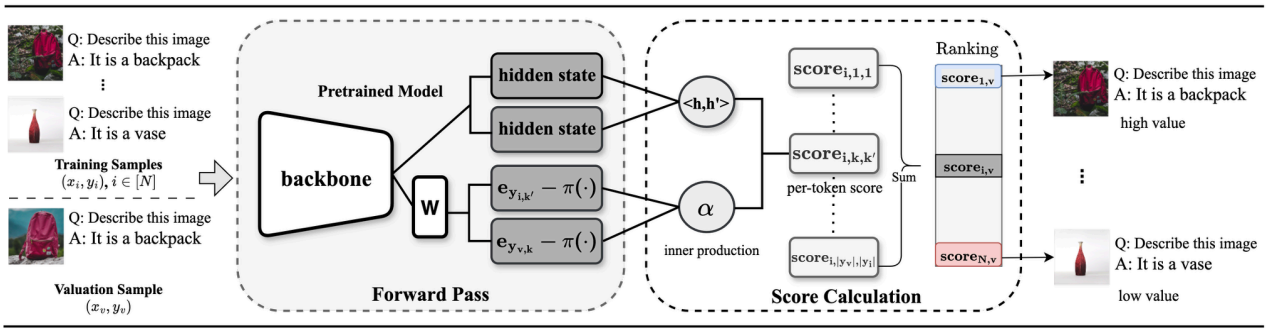


Figure 1: For-Value 的流程。给定一个估值样本和一个训练数据集，For-Value 对所有数据进行前向传播，计算每个训练样本的分数（Eq. (1)），使用最后一层隐藏嵌入和预测误差 α 。然后，基于这些计算得出的值对训练样本进行排序。

验证数据）中表现良好。表现良好的模型应该能够生成具有高置信度的验证数据。数据价值的概念也与困惑度度量密切相关，它反向反映了模型在生成给定文本时的不确定性。接下来，我们分析验证数据的对数可能性学习动态，$\frac{d}{dt}\ln \pi_{\theta(t)}(\boldsymbol{y}_v \mid \boldsymbol{x}_v)$ ，它反映了在增加生成验证输出的概率方面的学习目标。我们从以下假设开始：

Assumption 1 (Unconstrained Features). 具有足够表达能力的神经网络（例如，预训练的 LLMs/VLMs）可以生成独立于架构具体复杂性的无约束嵌入 $\boldsymbol{h}_{\boldsymbol{x}} \in \mathbb{R}^d$ 。这些嵌入随后通过一个令牌反嵌入矩阵 $\boldsymbol{W} \in \mathbb{R}^{|\mathcal{V}|\times d}$ 转换为 logits。生成的 logits 通过 softmax 函数进行处理，以在可能的下一个令牌上生成概率分布。为了给序列 $\boldsymbol{y} \in \mathcal{V}^*$ 赋予概率，语言模型 π_θ 以自回归方式运行，即，$\pi_\theta(\boldsymbol{y} \mid \boldsymbol{x}) = \prod_{k=1}^{|\boldsymbol{y}|} \mathrm{Softmax}(\boldsymbol{W}\boldsymbol{h}_{\boldsymbol{x},\boldsymbol{y}_{<k}})_{y_k}$.

在无约束特征设置下，训练样本对验证样本的影响表示为（详细证明见附录）：

Theorem 1. 对于一个样本 $\boldsymbol{x}_v$ 及其生成的 $\boldsymbol{y}_v$ 在等候评估时，在任何训练时间 $t \geq 0$ 使用训练样本 $(\boldsymbol{x}_i, \boldsymbol{y}_i), i \in [n]$ ，当没有训练输入 $\boldsymbol{x}_i$ 与评估输入 $\boldsymbol{x}_v$ [1] 相同时，随着以下增加，训练数据对评估数据表现出更大的价值：

$$\sum_{k=1}^{|\boldsymbol{y}_v|}\sum_{k'=1}^{|\boldsymbol{y}_i|} \alpha_{k,k'}(t) \cdot \left\langle \boldsymbol{h}_{\boldsymbol{x}_v,\boldsymbol{y}_{v,<k}}(t), \boldsymbol{h}_{\boldsymbol{x}_i,\boldsymbol{y}_{i,<k'}}(t) \right\rangle \tag{1}$$

其中 $\alpha_{k,k'}(t) =$

$$\left\langle \boldsymbol{e}_{\boldsymbol{y}_{v,k}} - \pi_{\theta(t)}(\cdot \mid \boldsymbol{x}_v, \boldsymbol{y}_{v,<k}), \boldsymbol{e}_{\boldsymbol{y}_{i,k'}} - \pi_{\theta(t)}(\cdot \mid \boldsymbol{x}_i, \boldsymbol{y}_{i,<k'}) \right\rangle$$

用于量化样本间的令牌级别预测误差的相似性。为简洁起见，后面部分我们省略 (t) 。

如定理所示，较大的得分对应于给定验证数据的可能性更大的增加，因此表示较高的估值。由于计算此得分仅需一次前向传播，我们将 Eq. (1) 表示为 For-Value 。

2.1 For-Value 的实现

引入我们的数据估值得分 For-Value 后，我们现在描述其用于可扩展实现的实际计算。Fig. 1 说明了我们方法的总体流程，下面提供了进一步的详细信息。

矩阵相似性：首先，我们将 (1) 重写成矩阵内积的形式。

$$\left\langle \sum_{k=1}^{|\boldsymbol{y}_v|} \left(\boldsymbol{e}_{\boldsymbol{y}_{v,k}} - \pi_\theta(\cdot|\boldsymbol{x}, \boldsymbol{y}_{v,<k})\right) \boldsymbol{h}^T_{\boldsymbol{x}_v,\boldsymbol{y}_{v,<k}}, \right.$$
$$\left. \sum_{k'=1}^{|\boldsymbol{y}_i|} \left(\boldsymbol{e}_{\boldsymbol{y}_{i,k'}} - \pi_\theta(\cdot|\boldsymbol{x}, \boldsymbol{y}_{i,<k'})\right) \boldsymbol{h}^T_{\boldsymbol{x}_i,\boldsymbol{y}_{i,<k'}} \right\rangle \tag{2}$$

重要的是，我们的重新表述涉及到在进行内积之前计算 k, k' 上的求和。此重新表述将整体复杂性降低到单一矩阵内积的程度。

Algorithm 1: 对于价值: 仅向前的数据估值

Input: Training set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$; valuation pair $(\boldsymbol{x}_v, \boldsymbol{y}_v)$; model π_θ ; batch size B .
Output: Data valuation $\mathcal{S}$.

Compute $\{\boldsymbol{h}_{\boldsymbol{x}_v,\boldsymbol{y}_{v,<k}}\}_{k=1}^{|\boldsymbol{y}_v|}$ and $\{\pi_\theta(\cdot|\boldsymbol{x}_v, \boldsymbol{y}_{v,<k})\}_{k=1}^{|\boldsymbol{y}_v|}$ by doing inference $\pi_\theta(\boldsymbol{x}_v, \boldsymbol{y}_v)$.
for each batch $\{(\boldsymbol{x}_j, \boldsymbol{y}_j)\}_{j=1}^B$ do
  Compute $\{\boldsymbol{h}_{\boldsymbol{x}_j,\boldsymbol{y}_{j,<k'}}\}_{k'=1}^{|\boldsymbol{y}_j|}$ and $\{\pi_\theta(\cdot|\boldsymbol{x}_j, \boldsymbol{y}_{j,<k'})\}_{k'=1}^{|\boldsymbol{y}_j|}$ by running batch inference.
  $\hat{\mathcal{V}} \leftarrow \bigcup_{j=1}^B \mathcal{V}_{\boldsymbol{x}_j,\boldsymbol{y}_j} \cup \mathcal{V}_{\boldsymbol{x}_v,\boldsymbol{y}_v}$
  Compute errors $(\boldsymbol{e} - \pi(\cdot))$ for tokens in $\hat{\mathcal{V}}$.
  For each in batch, compute $S_{v,j}$ via Eq. (2).
end for
$\mathcal{S} \leftarrow \{(\boldsymbol{x}_i, \boldsymbol{y}_i, S_{v,i})\}_{i=1}^N$.
Sort $\mathcal{S}$ by $S_{v,i}$ (descending).
return $\mathcal{S}$.

聚焦于已见过的词汇：该公式涉及计算预测误差向量（例如，$\boldsymbol{e}_{y_{i,k'}} - \pi_\theta(\cdot|\boldsymbol{x}, \boldsymbol{y}_{i,<k'})$ ）与隐藏嵌入之间的外积，这导致计算复杂度为 $O(|\mathcal{V}|d)$ 。由于概率质量主要集中在样本的词汇上，我们将计算限制在与样本词汇相关的词汇 $\mathcal{V}_\mathcal{D}$ 上。鉴于 $|\mathcal{V}_\mathcal{D}| \ll |\mathcal{V}|$ ，这显著降低了整体成本至 $O(|\mathcal{V}_\mathcal{D}|d)$ （详见 Tab. 3 中的效率比较）。值得注意的是，当进行每批次估值计算时，词汇表大小可以进一步减少到批内词汇大小，如算法 algorithm 1 的步骤 6 所示。

[1] 这个假设比较温和，因为在实际应用中，训练输入通常与评估输入不同——例如，在视觉-语言数据集中，输入图像几乎总是不同的。更多讨论见附录。

Method	Qwen2.5-1.5B		Llama-2-13B-chat	
	AUC ↑	Recall ↑	AUC ↑	Recall ↑
Sentence transformations				
Hessian-free(?)	0.785 ± 0.096	0.370 ± 0.139	0.999 ± 0.002	0.985 ± 0.033
DataInf(?)	0.981 ± 0.019	0.826 ± 0.121	$\mathbf{1.000 \pm 0.000}$	0.997 ± 0.010
HyperINF(?)	0.993 ± 0.013	0.934 ± 0.063	$\mathbf{1.000 \pm 0.000}$	0.998 ± 0.011
Emb(?)	0.546 ± 0.306	0.148 ± 0.205	0.854 ± 0.192	0.563 ± 0.412
For-Value	$\mathbf{1.000 \pm 0.001}$	$\mathbf{0.989 \pm 0.025}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.001}$
Math Problem (w/o reasoning)				
Hessian-free(?)	0.835 ± 0.235	0.592 ± 0.291	0.770 ± 0.174	0.258 ± 0.388
DataInf(?)	0.985 ± 0.032	0.878 ± 0.154	$\mathbf{1.000 \pm 0.000}$	0.999 ± 0.006
HyperINF(?)	0.986 ± 0.024	0.942 ± 0.080	0.995 ± 0.018	0.967 ± 0.057
Emb(?)	0.555 ± 0.298	0.146 ± 0.295	0.762 ± 0.239	0.389 ± 0.477
For-Value	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.998 \pm 0.011}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.002}$
Math Problem (w/ reasoning)				
Hessian-free(?)	0.829 ± 0.172	0.524 ± 0.350	0.772 ± 0.173	0.258 ± 0.388
DataInf(?)	0.987 ± 0.030	0.892 ± 0.155	1.000 ± 0.001	0.996 ± 0.025
HyperINF(?)	0.988 ± 0.023	0.950 ± 0.060	0.994 ± 0.018	0.961 ± 0.074
Emb(?)	0.560 ± 0.310	0.198 ± 0.311	0.725 ± 0.217	0.270 ± 0.420
For-Value	$\mathbf{1.000 \pm 0.000}$	$\mathbf{0.998 \pm 0.008}$	$\mathbf{1.000 \pm 0.000}$	$\mathbf{1.000 \pm 0.000}$

Table 1: 大型语言模型的影响数据识别性能。For-Value 在识别影响数据方面始终提供可比或更优的性能。

For-Value 算法：算法 1 总结了我们对 For-Value 的高效批次计算。我们首先通过对估值和训练批次进行单次前馈传递提取隐藏嵌入和预测误差。将计算限制在批内词汇并进行批处理计算显著减少了开销，同时保留了准确性。最后，我们对分数进行排序，以根据其估计影响对训练样本进行排名。重要的是，该算法可以通过对估值对的影响分数进行平均来自然地扩展到一组估值对。完整的流程在 Fig. 1 中有所说明。

3 实验

在本节中，我们描述了实验设置。附录中提供了更多细节。

基准方法。我们专注于与其他为效率设计的数据估值方法的比较：Hessian-free (??) 通过一阶梯度的点积估计影响分数，这相当于在最后的训练迭代中应用的 TracInf (?) 或一阶在运行中的 Shapley (?)。DataInf (?) 是一种为参数高效微调设计的数据估值方法，采用计算上高效的 Hessian 近似。HyperINF (?) 结合了一般的 Fisher 信息作为 Hessian 矩阵的低秩近似以提高精度。最后，我们采用最近一种基于嵌入相似性的方法 (?)，原本是为图像生成模型提出的，我们将其称为 Emb。值得注意的是，当应用于 LLMs/VLMs 时，该方法未考虑嵌入对对数似然变化的影响，也不包含预测误差。

模型。按照 (?)，我们使用 Llama-2-13B-chat (?) 和 Qwen-2.5-1.5B (?) 对 LLMs 进行评估，以涵盖更广泛的模型尺寸和家族。此外，由于我们方法的高效性，我们能够在 Qwen2.5 系列模型上运行 For-Value，参数从 7B 到 72B 不等。相比之下，基准方法需要大量训练和长时间运行，使其在这些更大模型上成本高昂。对于 VLMs，我们采用广泛使用的 Qwen-2.5-VL-3B-Instruct (?) 和 Llama-3.2-11B-Vision (?)。

重要数据识别任务。我们按照 (?) 对所有方法进行重要数据识别评估，主要是针对 LLMs 和 VLMs 进行。在 LLMs 方面，我们使用了三个文本生成数据集：(i) 句子转换，(ii) 不带推理的数学文字题，以及 (iii) 带推理的数学文字题。每个数据集包括 10 个类，代表不同的任务或背景，每个类有 90 个训练样本和 10 个验证样本。在 VLMs 方面，我们从 (?) 的文本到图像生成配置调整为图像到文本生成设置：(i) 风格生成（卡通，像素艺术，线条草图），在三个风格类中共有 600 个训练样本和 150 个测试图-文本对，以及 (ii) 使用 DreamBooth 数据集 (?) 的主题生成，包含 30 个主题类，每个主题有 3 个训练样本，其余样本用于验证。我们采用 (?) 中的两个评估指标：(i) AUC，它衡量数据值与伪标签之间的相关性（如果训练和验证样本属于同一类则分配 1，否则为 0），并在验证点上进行平均；(ii) 召回率，定义为共享与验证点同一类的最高排名的重要训练样本的比例。对于所有基线，我们报告具有最高 AUC 的检查点的结果，因为检查点之间的性能波动显著。附录 ?? 中提供了更多详细信息和数据集示例。

错误标记数据检测任务。我们使用 Kaggle 猫狗分类数据集 (?) 在 VLMs 上，检查基线方法和我们的 For-Value 的错误标记数据检测能力。具体来说，我们将数据集转换为问答任务，模板为 “图像中的动物是什么？它是 [label]”（见附录中的示例 Fig. 5）。然后，我们为狗和猫各选择前 400 张图像，将 50 % 的标签翻转以引

Method	Qwen2.5-VL-3B-Instruct		Llama-3.2-11B-vision	
	AUC ↑	Recall ↑	AUC ↑	Recall ↑
Image-to-text subject generation				
Hessian-free(?)	0.979 ± 0.038	0.738 ± 0.399	0.961 ± 0.093	0.765 ± 0.365
DataInf(?)	0.989 ± 0.024	0.836 ± 0.318	0.958 ± 0.119	0.797 ± 0.323
HyperINF(?)	0.988 ± 0.047	$\mathbf{0.902 \pm 0.220}$	0.993 ± 0.025	0.919 ± 0.186
Emb(?)	0.841 ± 0.189	0.206 ± 0.458	0.841 ± 0.189	0.206 ± 0.379
For-Value	$\mathbf{0.994 \pm 0.018}$	0.897 ± 0.287	$\mathbf{0.995 \pm 0.040}$	$\mathbf{0.985 \pm 0.068}$
Image-to-text style generation				
Hessian-free(?)	0.515 ± 0.096	0.799 ± 0.162	0.515 ± 0.079	0.824 ± 0.145
DataInf(?)	0.520 ± 0.094	0.760 ± 0.181	0.515 ± 0.174	0.785 ± 0.164
HyperINF(?)	0.516 ± 0.055	0.860 ± 0.103	0.490 ± 0.090	0.821 ± 0.137
Emb(?)	0.560 ± 0.310	0.198 ± 0.311	0.553 ± 0.294	0.340 ± 0.467
For-Value	$\mathbf{0.895 \pm 0.138}$	$\mathbf{0.916 \pm 0.153}$	$\mathbf{0.974 \pm 0.059}$	$\mathbf{0.997 \pm 0.013}$
Mislabeled Data Detection				
Hessian-free(?)	0.719 ± 0.098	0.760 ± 0.088	0.962 ± 0.019	0.955 ± 0.068
DataInf(?)	0.760 ± 0.088	0.901 ± 0.147	$\mathbf{1.000 \pm 0.000}$	1.000 ± 0.003
HyperINF(?)	0.770 ± 0.077	0.916 ± 0.128	1.000 ± 0.001	1.000 ± 0.006
Emb(?)	0.741 ± 0.061	0.533 ± 0.075	0.933 ± 0.044	0.996 ± 0.015
For-Value	$\mathbf{0.885 \pm 0.055}$	$\mathbf{0.999 \pm 0.010}$	0.995 ± 0.008	$\mathbf{1.000 \pm 0.000}$

Table 2: 不同 VLM 任务中关于识别有影响的数据和检测错误标记数据的性能。For-Value 在识别有影响的数据和检测错误标记数据方面，在各种 VLM 任务中始终提供了相当或更优的性能，与基准方法相比有更好的表现。

入噪音。为了验证，我们使用 200 张图像，每个类别包含 100 张图像。我们还报告了噪声数据识别的 AUC 和 Recall，详见附录。

效率评估。我们在 Qwen2.5 系列模型上对 For-Value 的运行时间进行了基准测试，模型规模从 1.5B 到 72B。对于 14B 以下的模型，我们使用单个 A100（80G）GPU；对于较大的 32B 和 72B 模型，进行推理时使用 4 个 A100 GPU，并用一个 A100 进行值计算。对于需要训练的基线方法，我们最多使用 8 个 GPU 进行微调，并将 32B 模型量化到 8-bit 精度，以便在一个 A100 上进行值计算以进行公平比较。由于基线的长运行时间，我们仅运行句子转换任务，并对于 14B/32B 模型，抽取 10 % 的验证数据——将时间缩放到 10 倍以估计总运行时间。

4 结果

在本节中，我们详细说明了 For-Value 和基线在 LLMs 和 VLMs 上的结果，并额外进行高效和定性分析。

4.1 影响力 & 是否标错数据识别

LLM 上关键数据识别结果。我们首先在 Tab. 1 中展示文本生成任务的结果，其中 For-Value 在所评估的 LLM 基准测试中始终与所有基线方法匹配或超越：

(1) 句子转换：如在 Tab. 1 中所示，对于句子转换任务，For-Value 在两个模型中都实现了完美或近乎完美的 AUC 和召回率。值得注意的是，在 Qwen2.5-1.5B 上，For-Value 在 AUC 上超越了最强的基线 HyperINF，提升了 0.7 %，并在召回率上有显著的 6.5 % 的提升。

(2) 数学问题（有 & 无推理）：在数学问题任务中，无论是否包含推理（数据样本于 Tab. 5 中），也表现出类似的模式。如在 Tab. 1 中所示，For-Value 仅通过一次前向传递就实现了更高质量的影响力识别，在两个 math 数据集中，使用 Qwen 模型比最佳的基线 HyperINF 提高了大约 6 % 的召回率。

这些结果表明 For-Value 能够可靠地识别不同任务和模型规模下的关键数据点，结合了较强的准确性和实际效率。

有影响的数据识别在 VLM 上的结果。接下来我们在 Tab. 2 中报告 VLMs 的结果。

(1) 对于主题生成，For-Value 在 Qwen-2.5-VL-3B-Instruct 和 Llama-3.2-11B-Vision 中都取得了最高的 AUC 和召回率，始终优于所有基线。具体而言，For-Value 在 11B 模型中召回率上超过最强的基线 HyperINF 超过 7 %。

(2) 在更具挑战性的风格生成任务中，For-Value 显示出明显的优势，与基线相比 AUC 提升超过 0.35，且相较于 Emb 方法有更大的增益。值得注意的是，基线的性能在此任务中下降得更为明显，这对其在复杂数据集上的鲁棒性提出了质疑。

这些发现证实了 For-Value 有效地识别了 VLMs 跨多样任务和模型大小的关键数据点。

错误标记数据检测。我们在 Tab. 2 中的错误标记数据检测结果展示了 For-Value 在不同模型规模下的强大性能。对于 Qwen-VL-3B 模型，For-Value 相较于最佳基

Method	Training Free	Algorithm Agnostic	Training Complexity	Computational Complexity	Memory Complexity
Original IF	✗	-	$O(nEd_{in}dL)$	$O(nd_{in}^2d^2L + d_{in}^3d^3L)$	$O(D^2L + nDL)$
Hessian-free	✗	✗	$O(nEd_{in}dL)$	$O(nd_{in}dL)$	$O(nd_{in}dL)$
DataInf	✗	✗	$O(nEd_{in}dL)$	$O(nd_{in}dL)$	$O(nd_{in}dL)$
HyperINF	✗	✗	$O(nEd_{in}dL)$	$O(nd^3L)$	$O(nd^2L)$
Emb	✓	✓	0	$O(nd)$	$O(nd)$
For-Value (ours)	✓	✓	0	$O(nd\|\hat{\mathcal{V}}\|)$	$O(nd\|\hat{\mathcal{V}}\|)$

Table 3: 影响函数（IF）、Hessian-free 、DataInf 、Emb 和 For-Value 的复杂性比较。假设一个多层感知器（MLP）有 L 个层，每层包含 $d_{in} \times d$ 个神经元，d_{in} 是输入维度，d 是输出嵌入维度，在 n 个训练样本上训练 E 个时期。不同层的参数数目相同（$D \in \mathbb{N}$），批内的词汇量大小为 $|\hat{\mathcal{V}}|$ 。总体而言，For-Value 比基线方法在计算效率和内存效率上都要高。

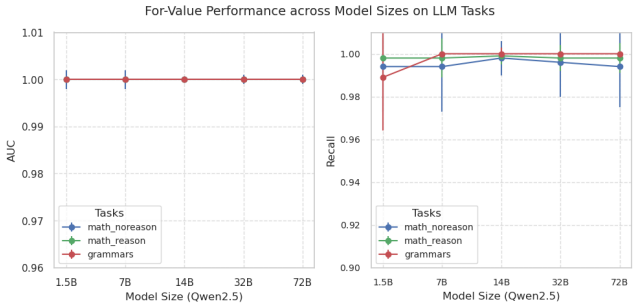


Figure 2: For-Value 在不同模型规模和任务上的表现。

线（HyperINF ）实现了高出 11.5 % 的 AUC 和 8.3 % 的召回率，显示了在识别错误标记样本方面的显著改善。该方法在更大的 Llama-3.2-11B 模型上同样表现出色，匹配了基于梯度的方法几乎完美的检测率（AUC $>$ 0.99，召回率 $= 1.0$）。这种在小型（3B）和大型（11B）VLMs 中的一致性能突显了 For-Value 的可扩展性和有效性。值得注意的是，For-Value 仅通过一次前向传递就达到了这些结果，仅需数秒，而基线方法则需几个小时。

4.2 消融研究 & 效率

预测误差相似性的影响 α 。我们通过将 α 设置为 1 来进行消融研究，以评估 α 项的作用，在 Eq. (2) 的计算中。这一简化将分数减少到 $\left\langle \sum_{k=1}^{|\boldsymbol{y}_v|} \boldsymbol{h}_{\boldsymbol{x}_v, \boldsymbol{y}_{v,<k}}, \sum_{k'=1}^{|\boldsymbol{y}_i|} \boldsymbol{h}_{\boldsymbol{x}_i, \boldsymbol{y}_{i,<k'}} \right\rangle$ ，该分数衡量两个数据样本间 $\boldsymbol{y}$ 的上下文化文本嵌入相似性（上下文是输入 $\boldsymbol{x}$ ，值得注意的是，在文本生成中，它是整篇文本，而在图像到文本生成中，它是文本部分）。这相当于 Emb 基线。如 Tab. 1 和 Tab. 2 所示，For-Value 在 LLM 和 VLM 任务中都始终显著优于 Emb 。这强调了在计算中包含 α 的重要性。直观地说，α 项中的预测误差起到了令牌级别权重的作用：当模型对训练数据中的一个令牌的信心水平已经很高时，其预测误差很小，贡献的梯度信号很少（损失较小）；同样，当验证令牌被大信心预测时，其概率的进一步增加受到限制，意味着它较少受到训练数据的影响。尽管 Emb 在生成图像模型（侧重于匹配分布相似性）的数据评估中表现良好，但其表现下降表明直接将其应用于自回归模型由于不同的教师强制目标而无效。

For-Value 模型规模对性能的影响。Fig. 2 显示 For-Value 在不同模型规模和任务中保持始终如一的高性能。对于所有任务，AUC 和 Recall 都接近 1.0 ，表明模型规模的扩大并未降低其有效性。这种稳定性证实了 For-Value 在扩展到更大模型时能够很好泛化，同时保留准确性，使其在各种 LLM 任务的实际部署中具有可靠性。

复杂性分析。Tab. 3 比较了不同方法的训练、计算和内存成本。传统方法，如 IF、Hessian-free 、HyperINF 和 DataInf ，依赖于梯度追踪或 Hessian 计算，其造成的高成本随着模型大小的增加而变得不理想。相比之下，Emb 和 For-Value 是不需要训练且与算法无关的，显著降低了开销。虽然 HyperINF 在准确性方面是最强的基线，但其三次复杂性使其不适用于大型 LLMs——例如 Qwen-32B 模型需要大约 6 小时（Fig. 4b ）。尽管 Emb 达到了最佳的运行时间效率，但其性能落后于其他方法，如 Tab. 1 和 Tab. 2 所示。我们的算法 For-Value 保持强劲性能的同时，效率也很高。由于 $|\hat{\mathcal{V}}|$ 通常较小（通常低于 2k），For-Value 达到了比基线更低的计算和内存成本。

时间成本分析。为了进一步展示效率，我们比较了不同模型大小和任务下 For-Value 与基线的时间成本。如 Fig. 4a 所示，尽管模型参数从 1.5B 增加到 72B，For-Value 始终保持低运行时间。对于所有任务，运行时间维持在几百秒以内，突显了其实用的可扩展性。相比较而言，如 Fig. 4b 所示，对于句子转换任务，基线方法需要显著更多的时间——以小时而非秒计。表现最好的基线 HyperINF ，尤其在处理更大模型时成本很高，对 32B 模型的运行时间长达 6 小时。这进一步强调了 For-Value 的效率优势，以最低的计算成本提供具有竞争力或更优的性能。

定性演示。除了定量结果之外，我们还展示了由 For-Value 识别的定性示例。Fig. 3 展示了一个目标估值样本及其最具和最不具影响力的训练样本，这些样本由 For-Value 排名。具体来说，For-Value 成功识别出高度相关的训练点——例如，在句子转换的单词反序任务中选择相同任务的样本，或在图像到文本的任务中匹配相同的主题或艺术风格。相反，最不具影响力的样本显然较不相关，并且通常在任务或内容上与目标估值数据完全不同。

在这项工作中，我们提出了 For-Value ，这是一种专为预训练的 LLM 和 VLM 设计的，仅向前的数据估值框架。通过仅依赖单次前向传递来估计每个样本的影响，For-Value 消除了与梯度和 Hessian 计算相关的计算瓶颈。我们的理论分析将 For-Value 基于自回归建模的学习动态，为其有效性提供了坚实的基础。在各种任务和模型规模上的实证评估表明，For-Value 不仅能够

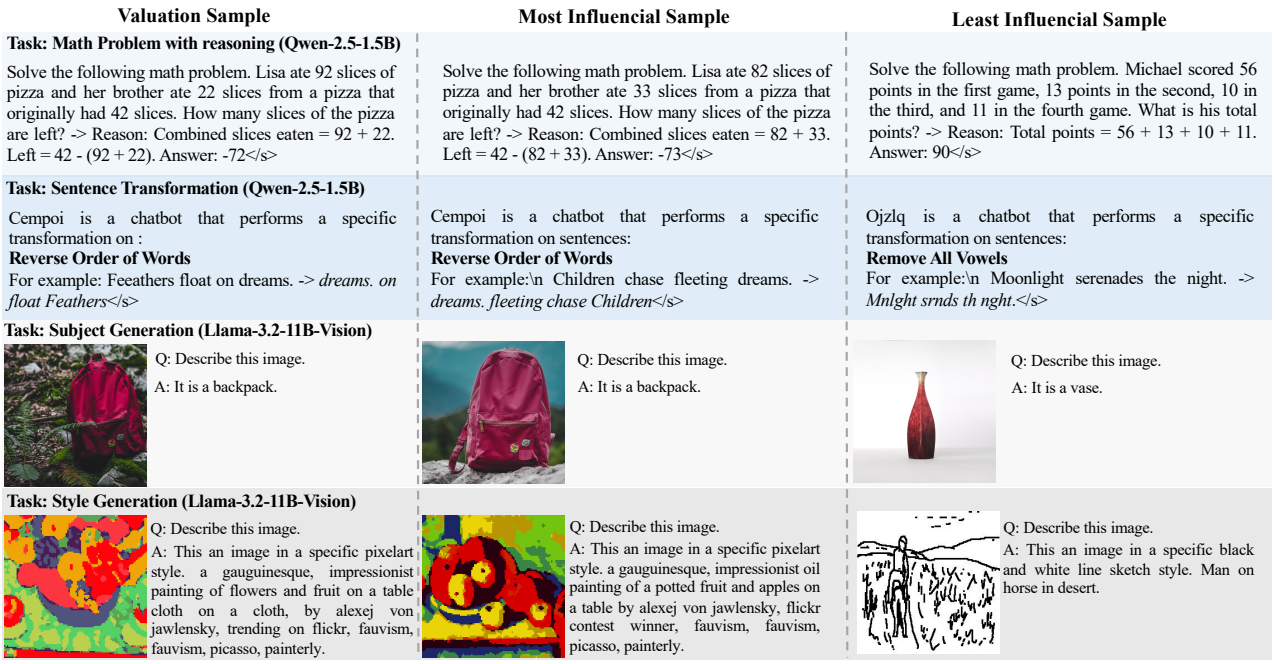
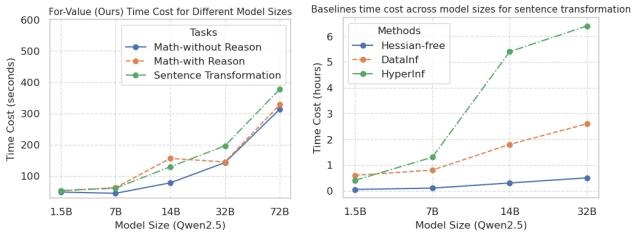


Figure 3: For-Value 识别的数据影响的定性示例。对于每个目标估值样本（左列），显示最具影响力的（中列）和最不具影响力的（右列）训练样本。For-Value 能够正确检索具有相关任务特征（例如，相同的推理类型、句子转换规则、主题或风格）的训练样本，并过滤掉不相关或不匹配的示例。



(a) For-Value 。　　(b) 基线方法

Figure 4: 时间成本分析：(a) For-Value 在不同模型规模和任务中的时间成本。(b) 不同模型规模下基线方法在句子转换任务中的时间成本。值得注意的是，For-Value 比基线方法显著更高效，时间成本以秒计算，而基线方法则需要长达数小时。

匹配基线的准确性，甚至在识别标签错误的数据点和检索最具影响力的训练样本方面常常超越基线，同时在计算效率上有显著提升。

为了将预训练的 LLM 或 VLM 适应特定领域或任务，模型通常在输入输出对的监督数据集 $\mathcal{D} = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^n$ 上进行训练。训练通常使用标准的教师强制目标进行，该目标最小化目标序列的负对数似然：

这一目标最大化模型在每一步生成正确输出序列的似然性，该输出序列以输入和真实前缀为条件。参数使用梯度下降或其变体进行更新：

其中 $\eta > 0$ 是学习率。教师强制通过在训练过程中提供真实前缀 $\boldsymbol{y}_{<k}$ ，稳定模型的微调，使得模型的预测能够与新领域的目标数据分布紧密对齐。

在本节中，我们提供了 ?? 的详细证明，我们首先证明以下定理：

Theorem 2. 对于尚待估值的数据 $\boldsymbol{x}_v$ 及其生成 $\boldsymbol{y}_v$ ，在使用训练数据 $(\boldsymbol{x}_i, \boldsymbol{y}_i), i \in [n]$ 的任何训练时刻 $t \geq 0$ ，除了依赖于标记的反嵌入，训练数据对估值数据表现出更大的价值，随着以下情况的增加：

$$\sum_{k=1}^{|\boldsymbol{y}_v|} \sum_{k'=1}^{|\boldsymbol{y}_i|} \alpha_{k,k'}(t) \cdot \left\langle \boldsymbol{h}_{\boldsymbol{x}_v, \boldsymbol{y}_{v,<k}}(t), \boldsymbol{h}_{\boldsymbol{x}_i, \boldsymbol{y}_{i,<k'}}(t) \right\rangle \tag{3}$$

我们观察到以下几点：

(1) 当训练输入 $\boldsymbol{x}_i$ 与估值输入 $\boldsymbol{x}_v$ 不同，其对估值目标的影响仅仅通过项 (I) 出现，项 (I) 捕捉了除了 token 解嵌入层之外的 token 嵌入和所有网络参数的贡献。

(2) 令牌卸载的影响集中在训练和评估数据共享相同输入 $\boldsymbol{x}$ 并展示出重叠输出预测 $\boldsymbol{y}$ 的情况下。为了消除对令牌卸载的这种依赖性，我们提出以下假设：

Assumption 2 (Distinct Input). 训练数据集满足以下条件：没有训练输入 $\boldsymbol{x}_i$ 与估值输入 $\boldsymbol{x}_v$ 相同。

在假设 2 下，token 反嵌入的贡献（项 (II)）消失，因此训练数据对估值数据的影响完全来自项 (I) 所捕获的共享表示特征。这一假设是温和的，因为在实际操作中，训练输入通常与估值输入不同——尤其是在视图-语言数据集中，输入图像几乎总是不同。将这个结果扩展到训练样本共享相同输入但输出不同的情况是简单的：输出前缀可以被并入输入中，将每一个唯一对待为一个不同的输入，其中表示输出开始不同的点。然后结合和假设 2 得出。根据，我们评估在三个大语言模型（LLMs）的文本生成任务中的表现，以识别有影响的数据点：

对于视觉语言模型（VLMs），我们将 (?) 的文本到图像生成任务改编为图像到文本（字幕）任务，以评估影响：

我们采用 (?) 中的两个指标来评估影响：

对于错误标记的检测，我们将数据集转换为视觉语言问答任务，使用模板 “图像中的动物是什么？它是一个 [label]”，并在 Fig. 5 中进行示范。然后我们选择 400 张狗和猫的图像，翻转 50 % 的标签以引入噪声。对于验证，我们使用 200 张图像，每个类别包含 100 张图像。对于评估，我们也计算 AUC 和召回率，但使用伪标签作为训练点，当训练点的标签与测试点相符且数据是干净的时为 1，否则为 0。

对于基准方法，我们选择具有最高测试 AUC 的模型检查点，因为基于影响函数的方法在训练检查点上表现出显著的性能变异性。值得注意的是，这种变异性与验证损失无关，这给实际应用带来了挑战。我们将 For-Value 与这些基准进行比较，以确保稳健的评估。

4.3 附加细节

基线的训练设置。虽然 For-Value 只需要一次前向传播，但基于影响函数的基线 Hessian-free 和 DataInf 需要微调模型直到收敛。对于文本生成任务，我们遵循 (?) 中的训练设置，但对于 llama-2-13B，我们使用 float16 权重而不是 8 位量化。对于图像到文本生成任务，我们在模型的注意力层中对每个查询和值矩阵应用 LoRA。为了微调 VLMs，我们使用学习率 2×10^{-4} ，LoRA 超参数 $r = 8$ 和 $\alpha = 32$ ，float16 模型权重，批量大小为 32，并训练 20 个轮次。

Table 4: 句子转换任务模板的描述。我们考虑了 10 种不同类型的句子转换。对于每种句子转换，独特的标识 “聊天机器人” 名称额外被添加到任务提示前，以帮助模型进行训练。

Sentence transformations	Example transformation of “Sunrises herald hopeful tomorrows.”
Reverse Order of Words	tomorrows. hopeful herald Sunrises
Capitalize Every Other Letter	sUnRiSeS hErAlD hOpEfUl tOmOrRoWs.
Insert Number 1 Between Every Word	Sunrises 1herald 1hopeful 1tomorrows.
Replace Vowels with *	S*nr*s*s h*r*ld h*p*f*l t*m*rr*ws.
Double Every Consonant	SSunrriisseess hheerraalldd hhopeefull ttomorrows.
Capitalize Every Word	Sunrises Herald Hopeful Tomorrows.
Remove All Vowels	Snrss hrld hpfl tmrrws.
Add ’ly’ To End of Each Word	Sunrisesly heraldly hopefully tomorrows.ly
Remove All Consonants	uie ea oeu ooo.
Repeat Each Word Twice	Sunrises Sunrises herald herald hopeful hopeful tomorrows. tomorrows.

Table 5: 数学问题任务模板的描述。我们考虑了 10 种不同类型的数学应用题。

Math Word Problems	Template prompt question
Remaining pizza slices	Lisa ate A slices of pizza and her br... had C slices. How many slices of ... eaten = A + B. Left = C - (A + ...
Chaperones needed for trip	For every A students going on a fie... ones. If C students are attending, h... needed = (B * C) // A.
Total number after purchase	In an aquarium, there are A shar... sharks, how many sharks would be... C.
Total game points	Michael scored A points in the first ... and D in the fourth game. What is ... + B + C + D.
Total reading hours	Emily reads for A hours each day. ... B days? Reason: Total hours read ...
Shirt cost after discount	A shirt costs A. There’s a B-dollar ... the discount?:Reason: Cost after d...
Area of a garden	A rectangular garden has a length ... is its area? Reason: Area = A * B.
Total savings	If Jake saves A each week, how mu... savings = A * B.
Number of full cake boxes	A bakery sells cupcakes in boxes ... boxes can they fill? Reason: Boxes ...
Interest earned	John invests A at an annual intere... earn after C years? Reason: Interes...

4.4 许可说明

Dreambooth 图像要么由论文作者拍摄，要么从 Unsplash [2] 获取。位于此链接 [3] 的文件包括 Unsplash 上所有图片的参考链接列表，连同摄影师署名和图片许可。草图图像来源于 FS-COCO (?) 。数据出处和图片许可可以在以下链接提供的文件中找到 [4] 。

[2] https://www.unsplash.com/

[3] https://huggingface.co/datasets/google/dreambooth/blob/main/dataset/references_and_licenses.txt

[4] https://github.com/pinakinathc/fscoco

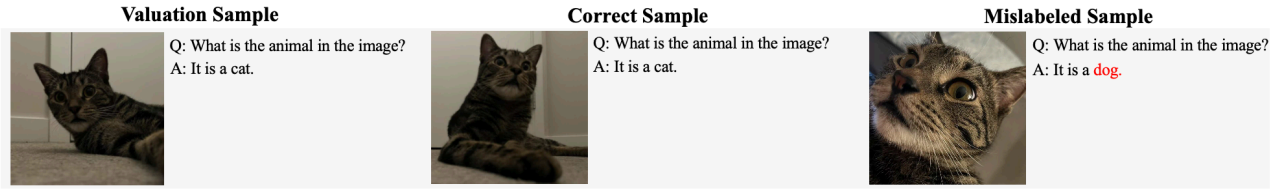


Figure 5: 误标数据检测任务的描述。我们使用猫和狗分类数据集，并通过随机交换 50 % 的数据标签有意引入噪声。